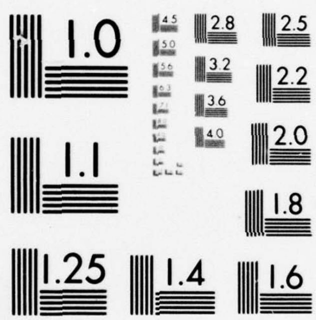
AD-A052 861 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/2
THE APPLICATION OF INEXPENSIVE MINICOMPUTERS TO INFORMATION WOR--ETC(U)
MAR 78

UNCLASSIFIED AGARD-LS-92 NL

1 OF 1
AD A052861

END
DATE FILMED
5-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AGARD-LS-92

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

See back page for 1473

AGARD LECTURE SERIES No. 92

The Application of Inexpensive Minicomputers to Information Work

NORTH ATLANTIC TREATY ORGANIZATION

DISTRIBUTION AND AVAILABILITY
ON BACK COVER

1

AGARD-LS-92

NORTH ATLANTIC TREATY ORGANIZATION

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT

(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Lecture Series No.92

THE APPLICATION OF INEXPENSIVE MINICOMPUTERS TO INFORMATION WORK

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD, presented on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

THE MISSION OF AGARD

The mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Exchanging of scientific and technical information;
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Providing scientific and technical advice and assistance to the North Atlantic Military Committee in the field of aerospace research and development;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Program and the Aerospace Applications Studies Program. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published March 1978

ISBN 92-835-1276-6

Printed by Technical Editing and Reproduction Ltd
Harford House, 7–9 Charlotte St, London, W1P 1HD

THEME

This Lecture Series No.92, on the subject of The Application of Inexpensive Minicomputers to Information Work, is sponsored by the Technical Information Panel of AGARD and is implemented by the Consultant and Exchange Programme.

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

This Lecture Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI and retrospective retrieval and library housekeeping.

LIST OF SPEAKERS

Lecture Series Director: Dr R.A.McIvor
Director
Scientific Information Services
Department of National Defence
Ottawa, Ontario K1A OK2
Canada

Dr J.H.Ashford
Lipman Management Resources Ltd
54–70 Moorbridge Road
Maidenhead SL6 8BN
U.K.

Dr C.J.Aslin
University of East Anglia
The Library
University Plain
Norwich NR4 7TJ
U.K.

Mrs A.N.Grosch
University of Minnesota
Wilson Library
Acquisition Division
Minneapolis, Mi 55455
U.S.A.

Mr G.W.Hart
Defence Research Information Centre
Station Square House
St Mary Cray
Orpington, Kent BR5 3RE
U.K.

Mr F.V.Wagner
Senior Vice President
Informatics Inc
21031 Ventura Boulevard
Woodlands Hills, California 91364
U.S.A.

Miss F.A.Daneliuk
IDRC
P.O.Box 8500
Ottawa, Ontario K1G 3H9
Canada

CONTENTS

USE OF MINICOMPUTERS IN DSIS

By
R.A. McIvor
Director Scientific Information Services
National Defence Headquarters
Ottawa, Ontario
K1A OK2

SUMMARY

A general introduction to the lecture series is given. The reasons for the choice of minicomputer by the Canadian Defence Scientific Information Service (DSIS) are discussed, and a general outline of the system is given. Particular attention is given to the data-input system and the advantages and disadvantages of various options are discussed. DSIS experiences with a variety of input methods are described. Some expected future developments are indicated.

Welcome to Lecture Series No. 92 on the Use of Inexpensive Minicomputers in Library Applications. Because we are presenting this series in more than one location, we have not been able to tailor the series to the needs of a particular country, but we hope the time set aside for discussion will be sufficient to allow problems of particular interest to your country to be discussed.

No definition was provided to the panel on what constitutes an "inexpensive" minicomputer. I have taken it to be "inexpensive" in comparison with medium or large-size computer systems rather than in comparison with the budget of a small library. I imagine the systems to be described would fall in general into the U.S. $50,000 - $100,000 range. The price of electronic hardware is still falling rapidly in proportion to other costs such as those of hiring staff, so that smaller and smaller libraries may find automation cost-effective. In North America now, quite respectable micro-computer systems may be obtained for U.S. $4,000-$5,000. Such systems would have two flexible disc drives on-line and 32,000 characters of random-access memory. Such equipment with this capability could handle a small information system. I would not like to give the impression that it would be desirable to purchase such equipment for this application at the present time. Many of the components are produced by small companies whose customer service leaves much to be desired, and the requisite software packages are not available. The cost of developing software for a microcomputer system for such a specialized application would far exceed the cost of the hardware. Nevertheless the trend is evident. Towards the end of 1977 Radio Shack, a chain of stores selling electronic equipment throughout North America and to a lesser extent elsewhere, introduced a packaged ready-to-operate microcomputer with keyboard, processor, memory, display monitor and cassette-tape recorder for storing programmes, all for about $600. U.S. They plan to introduce before long a low-cost printer, floppy disc units and software for personal and business applications.

In our Lecture Series, I have chosen to have various aspects of library automation discussed in individual lectures beginning with "Choosing the Computer" through to "Future Prospects in Minicomputers" including such topics as cataloguing, circulation, selective dissemination of information, preparation of an abstracts journal etc. One subject not specifically mentioned is data input. I therefore thought for my introductory lecture, I would give briefly an overview of our present automated system at the Canadian Department of National Defence Scientific Information Service, with emphasis perhaps on the data input aspects.

First of all, it might be worthwhile to spend a few minutes discussing - Why did we choose a minicomputer? Many of us will recall Grosch's law, which has been stated in many forms but the usual interpretation is that the larger, and hence more costly the computer, the more cost-effective it is, assuming it is used to full capacity. Grosch's law certainly seemed to apply several years ago to the products of a particular company, but cynics have seen in the law more a reflection of the marketing policies of that company rather than a law of nature. However, out of this has grown the mystique of the large computer enshrined in its Holy of Holies surrounded by high priests and acolytes scrambling to keep it adequately fed 24 hours a day with a penalty of perhaps .15¢ per idle second. At DSIS we operated on an IBM 360/65 for 4 years. It was a system much as I described. Because of the nature of our data base to which a restricted access must be enforced, we had to dedicate this machine entirely to ourselves when we used it. For reasons of economy and scheduling, we had one or two hours per week, generally just before or after midnight. In the early stages, programmes were still not error-free and we would sometimes find that our work had been wasted, and that another week had to elapse before we would have another chance.

During this time we were building up experience with our minicomputer which had been acquired for data-input, transferring some of the tasks to it. We were pleased at the convenience of having things done when we wanted, and found that the cost of the in-house processing was 2 to 10-fold less than for the same programme run on the larger system. In the meantime prices had come down and we had enough information to show that transferring our entire processing to the minicomputer would be cost-effective. A second minicomputer system was therefore acquired for this purpose, and the first system was given the same capability as the second so that in emergencies one system could do all the work by using a second shift if necessary.

Aside from the likely overall saving in costs, the lower costs of enhancements such as additional memory, peripheral equipment etc. make it easier to upgrade the system in subsequent years.

However, a factor that cannot be overemphasized is the convenience of having a dedicated in-house system to do the jobs that are wanted when they are wanted. The computer then becomes the servant rather than the master. A recent Datamation article estimated that if the computer applications in an organization could occupy at least 20% of the available time of a minicomputer it was probably worth obtaining one. Certainly we have not regretted our decision in 1972 to transfer our operations to an in-house minicomputer.

To return to the DSIS automated system, a brief history of the project is perhaps in order. The planning and groundwork for the automation of the DSIS system was done in the period 1955-56. A 5-year program was laid out, nicknamed SOCRATES - System of Organization of Current Reports to Aid Technologists, and Scientists. The plan was to phase it in parallel with the manual system. We are proud and pleased that the programme was completed on time, within budget and without an increase in staff. At that time, the plan was to use a large or medium sized computer for the data processing, with a minicomputer for data-input, and it was envisaged that the minicomputer could later serve as a front-end processor for an on-line system.

In implementing any automated system, it is important to introduce changes gradually, and always have the capability to backtrack when the almost inevitable failures or delays occur. Not only will the staff be less disturbed by the changes, but production can continue reasonably on schedule. In the main, we followed this plan although we had many problems and delays, we had only one severe disruption when we were obliged to make a major change in our input system without an opportunity for parallel operation during the changeover.

When the original minicomputer was obtained in 1969 for some $25,000. and placed in the data input role, further experimentation showed that with some increase in its capacity, addition of disc, tape unit and printer, it could be made capable of handling all aspects of our processing, except the on-line retrospective search system. By 1973, another $25,000 of equipment had been added and a second $50,000 system had been acquired. At that time all work was transferred to in-house minicomputers except for our KWOC Index which it was considered uneconomic to transfer.. By 1976 a third $75,000 system with large capacity disc had been added. The original mini has been retired, and the second is being phased out. In 1978 we expect to duplicate the third system in preparation for transferring all remaining operations, including the retrospective search system, in-house. Our hardware then will consist of 2 Sperry-Univac (formerly Varian) Series 70 computers, each with some 100K bytes of memory, two tape drives and a 96M byte disc.

To return to the data input role, it is clear that no automated system can be introduced without getting the data into machine readable form. There are several approaches to this: (TABLE 1) Each of these has some advantages and disadvantages which are summarized in TABLE 2 - 6.

At DSIS we have had experience with a number of data-input methods. Punched cards were ruled out for reasons of limited character set and fixed field structure. We used Flexo-writers (paper-tape) for a couple of years, but abandoned them because of the noise, slow playback and unreliability of the paper-tape to magnetic-tape conversion equipment. Subsequently, a minicomputer with one tape drive and 4,000 16-bit words of memory was used in a time-sharing mode with 4 input typewriters. While this equipment served the purpose, it was soon realized that a better, more reliable system would require 8,000 words of memory and two tape-drives, and that a printer was needed to provide "instant playbacks" for proofreading, as the original plan to use the input terminals for playback was unsatisfactory.

The most recent stage in our data-input system has been the return to stand-alone terminals. Our typewriter-terminal supplier went out of business, and the original minicomputer, now eight years old was having more and more down time. As mentioned earlier all time-shared input is brought to a halt by a failure in the central processing unit. A shift to visual display units at this time speeded up input two-fold, by simplifying input procedures but caused considerable disruption for some months before an adequate replacement was developed for the three-part card produced by the typewriter terminals.

Our first mistake in implementing data-input was to purchase typewriter terminals, combined with custom-built interfaces between the computer, tape drive and terminals. Design faults in the interface led to frequent breakdown of the terminals, the company that built the interfaces went bankrupt, and the terminal equipment was soon obsolete. Finding anyone to service the equipment soon became very difficult, and we had to phase out some of the equipment before it would normally have been done. When this replacement became necessary, we leased the new terminals. Service is generally excellent on leased terminals and they are exchanged when they cannot be repaired quickly on-site. Also, if improved equipment becomes available, it is often possible to renegotiate the lease, or at least exchange the equipment when the lease comes up for renewal.

A second error was acquiring only one tape drive initially. Peripheral equipment usually gives more trouble than the computer, and failure of the tape drive meant all input was stopped until it was repaired. Secondly, it is usually more convenient to attempt to obtain a clean input file before updating, rather than to correct the master file later. Two tape drives allow editing by tape-to-tape copying, since a single magnetic tape cannot be reliably corrected by overwriting.

A third error was not obtaining in the first instance tapes with read-after-write capability. The inexpensive models usually sold with minicomputers do not generally have this capability. While read errors in the input file are inconvenient, they are not catastrophic, but in maintaining a master file the tapes are copied very frequently in updating and report generation. Errors not detected during this process accumulate and require a great deal of time and effort to correct. Automatic read-after-write ensures that records written on a bad spot on the tape, or corrected by some electronic glitch are immediately detected, and can be rewritten on a fresh section of tape.

In our manual system, descriptive cataloguing data were entered on a three-part card. The top copy was used as a work card to accompany the document during processing, the second served as a temporary catalogue card, and the third, on stiffer material, was filed as a loan-control card. In automating the data input, upper and lower case typewriter terminals were chosen so that this three-part card was preserved. Our only major disruption resulted from the loss of this card when a shift was made to visual display units. As mentioned, we did not have adequate preparation time for this change, the units were delivered late and did not match their specifications exactly. For some months we had to type these cards manually before the programme was operable for producing them automatically from the floppy-disc record.

I hope with this brief outline of our experiences with data input to have given some indication of the advantages and disadvantages of various methods, and one centre's experiences with them.

Once the data is input, it must be incorporated into some sort of master file. This process can be done either by batch, or on-line, but in the small minicomputers is more likely to be done by batch. The master is kept on magnetic tape, new additions and modifications are accumulated until a sufficient number are available, usually for a week to a month in a small centre. The master tape is then copied onto a new file incorporating additions and modifications, taking care to preserve the old files and the corresponding transaction files containing the additions, deletions and modifications, for a sufficient time to ensure that any errors in the file caused by program modifications or undetected program faults have been detected. I cannot over emphasize the value of thorough checking of all outputs at each stage for a considerable period of time after any change is made to a programme to catch errors before the file is corrupted past recovery.

There are various degrees of sophistication possible in the design of a master file for a bibliographic data base, and at DSIS we are now developing our fourth. Our first one, developed in 1967, was a simple one - easy to implement, but inefficient in processing time for some applications. Essentially it consisted of a series of variable length fields, each with a header identifying it, and a special delimiting character indicating the end. Subsequent designs introduced a directory which facilitated the rapid skipping of fields unwanted for a particular application, and extensions on the types of field-subfield structures permitted. Eventually, our five or six data bases ended up in one of three major data base structures with slight irregularities even within a single structure. This made it necessary to have sets of almost identical programmes to process them. This proliferation of programmes and our near future switch to a new computer with a different operating system were the incentives to consolidate our data bases into a new unified data base structure.

The new data base structure was designed with two major goals in mind. First, we wanted to be able to process any of our data bases with any programme. This was achieved by associating with each data base a simple data dictionary. The data dictionary permits programmes such as verification to be identical across all the data bases. In addition, each master file is preceded by a subset of the data dictionary which contains a mapping between the access name for a field and its location in records in the master file. This permits programmes, such as report generators, to run independent of the data base, and enables us to modify a data base's data dictionary without worrying about rendering earlier master files inaccessible. The second major goal of our new data base structure was to be flexible enough to quickly respond to requests for new data bases. This second goal was achieved through the data dictionary concept, and through the power of our internal structure. The data dictionary enables us to rapidly describe a new data base to our programmes. The internal record structure, a generalized hierarchy, permits us to intuitively model a large number of data bases. Of course, the generalization of our programmes and the more powerful internal record structure are not free. We will require more processing time and more storage space for our new system but we are willing to accept these trade-offs for the convenience of a more flexible system.

As many automatic checks as can be conveniently implemented with the equipment available should be incorporated in the master file updating system. With our current system some of these checks are done at the data input stage (e.g. field length, whether the field is alphabetic or numeric, upper and lower case, size range, specific values from a short list etc) while others are done at the updating stage (validation of subject codes, thesaurus terms, and automatic insertion of broader terms from the thesaurus). Lack of disc space has in the past prevented us from automating our corporate source authority file, but we plan to do so when sufficient effort is available.

Each time the master file is updated, it is customary to select new or changed records for periodic report production - such things as SDI which I will be describing in more detail later; preparation of an abstracts journal, which will be dealt with by Mr. Hart, and possibly such additional products as catalogue cards or computer output microfilm with indexes for retrospective searching, or items for incorporation into an on-line computerized retrospective search system. We have done all of these things at DSIS, but some years ago dispensed with catalogue cards in favour of microfilm cartridges.

At DSIS we have never considered it cost-effective to computerize our loan-control or circulation records. However, for many libraries this is a valuable service that can be performed on minicomputers and will be dealt with in detail by Dr. Aslin.

While our DSIS on-line enquiry system is still based on a larger computer, our long-term plans are to place it on a minicomputer system as well. In this respect, our colleagues at the International Development Research Centre are further advanced, and Ms. Faye Daneliuk will be describing their system later in this series.

I hope I have achieved my objective of giving a very brief overview of the minicomputer automated system as it is currently in operation at DSIS as well as an indication of alternative methods with some of their advantages and disadvantages.

TABLE 1
POSSIBLE DATA-ENTRY SYSTEMS

1) Punched cards
2) Paper tape
3) Key-to-tape
4) Key-to-disc - Time sharing systems
5) Floppy disc, cassette or magnetic card terminal
6) OCR (Optical Character Recognition)

TABLE 2
PUNCHED CARDS

ADVANTAGES

1) Equipment readily available and relatively inexpensive
2) Data readily verified and edited

DISADVANTAGES

1) Lower case letters not usually readily available
2) More adapted to fixed format applications
3) Output useful only for computer entry

TABLE 3
PAPER TAPE TYPEWRITER

ADVANTAGES

1) Each unit independent
2) Output less bulky than cards
3) Upper and lower case available
4) Output a useful by-product

DISADVANTAGES

1) Noisy
2) Not readily edited
3) Slow to playback
4) High speed handling equipment going out of fashion

TABLE 4
KEY-TO-TAPE (TIME SHARING)

ADVANTAGES

1) Material produced in immediately suitable form
2) Readily edited

DISADVANTAGES

1) When control processor or tape unit is down, all machines are idle
2) Equipment more expensive
3) If CRT
VDU - no hard copy

TABLE 5
FLOPPY DISC

ADVANTAGES

1) Each unit independent
2) Formatted screens and preprocessing
3) Ready editing

DISADVANTAGES

1) Expense
2) Require more skilled operators

TABLE 6
O C R
(OPTICAL CHARACTER RECOGNITION)

ADVANTAGES

1) No keyboarding required

DISADVANTAGES

1) Expensive
2) Relatively high error rate
3) Copy must be high quality and/or typed with special font

SELECTION OF MINICOMPUTER SYSTEMS FOR BIBLIOGRAPHIC APPLICATIONS

Audrey N. Grosch
Associate Professor / Special Projects Officer
University of Minnesota Libraries
Minneapolis, MN 55455 U.S.A.

SUMMARY

This paper considers the selection of minicomputer systems for a wide range of potential bibliographic applications -- from simple single dedicated library tasks such as circulation control through complex integrated library management or interactive retrieval systems. Basic assumptions and definition of a given system's capabilities, required functions, method of creation including design and development, and working environment are introduced. This framework is necessary prior to determining whether distributive processing employing either stand-alone or linked minicomputers is desirable or whether a more traditional conventional shared large-scale computer system is in order. Minicomputer system development trends are highlighted to set in perspective a discussion of criteria for system selection. These criteria are of two categories. The first relates to the definition of the application. The second evolves from the first category. It is the determination of hardware, software and general system factors having prime importance in system selection for bibliographic purposes. The minicomputer central processor, main memory, memory protection, peripheral devices, data communication interfaces, vendor supplied software, and instruction sets are discussed in light of bibliographic applications. Other evaluative criteria such as vendor support, delivery and pricing schedules and multiple vendor systems are briefly considered.

INTRODUCTION

Minicomputer systems, although physically smaller and consuming less electrical power than conventional mainframe computers, possess more and more of the qualities of their larger relatives. The minicomputer itself is really a modular central processor belonging to a highly modular family of devices. To the user, this means that through a wise choice of host system for his specified tasks, a gradual growth of the hardware and software system can occur as specific need indicates.

Certain assumptions must be thought out by the user prior to selection of a system. Moreover, the user should understand the trade-offs in use of distributive processing employing these modular minicomputer systems to those of shared, large-scale conventional computer systems. Finally, it is helpful to understand the trends in minicomputer development as these impact on a selection of any host computer system for any long term system commitment.

In an earlier paper,[1] this author explored the economics and selection of minicomputer systems for integrated library management systems. In this subsequent paper a more mature environment will be considered and bibliographic applications of all types can be served by the points addressed here in the light of a rapidly advancing hardware and software technology.

BASIC ASSUMPTIONS

Prior to any equipment or software selection, a broad definition of the application and its environment should be completed. This broad definition should explain the tasks the system is to perform, the specific data to be used and stored by the system, the response requirements of the system in its online operations for data entry, editing, updating, and inquiry. The size of the data base initially, its growth rate, and the number and geographic location of terminal users should be determined. The features and capabilities which are absolutely required should be a key part of this definition.

With a description of the application system and its environment in hand, the next assumption that should be made is, "How will this system be brought to an operational state from its conceptualization?" A number of alternatives may be possible such as:

- in-house development of systems and application software using hardware selected and configured for the system;
- in-house development of applications using data base management systems software, operating system and other utilities software selected along with an appropriate hardware configuration;
- contractor development of systems and/or application software according to specifications for a hardware system chosen in-house; or
- a turn-key system procured complete and ready to operate from a specialist in such systems according to a detailed procurement specification.

Which of these alternatives will be employed to bring a given system to an operational state will depend upon the agency's in-house programming and systems capabilities, available personnel, and accepted procedural practices. However, if contractors are to perform all functions, it is still important that the agency personnel know what is involved in choosing a host hardware/software system so that proper contractor evaluation, selection and monitoring may be done as the project proceeds. Thus, contract milestones may be evaluated with a degree of confidence that prior decisions were as sound as possible based on current knowledge and technological capabilities.

DISTRIBUTIVE PROCESSING VS. SHARED LARGE-SCALE COMPUTER SYSTEMS

Ten years ago virtually anyone contemplating design of a bibliographic system -- whether a system to perform typical library business functions such as material ordering, accounting, serials management, circulation or cataloging or a very elegant retrieval system for complex structure searching of very large formula or text files -- had only one alternative in a host computer. This was the shared or dedicated use of a large-scale computer system which usually had been procured with quite different applications in mind and different users dictating the priorities of the system. All too often in such systems, a bibliographic application of any size did not fit well into the job stream of the system. The reasons were that bibliographic systems contended for large blocks of memory, large numbers of disk accesses stealing machine cycles, and generally a large amount of channel time for peripheral device servicing. This dependence on input/output task completion rather than purely compute-bound tasks strains the job queue management for systems already performing predominantly non-computational applications. Moreover, bibliographic applications wasted the main resource of the large-scale computer, its speed, in order to get the large memory, telecommunications, and other peripheral support such as high capacity mass storage.

Today, a more realistic choice of host computer can occur which can improve the user's control and responsiveness of the system, conserve valuable computing resources not needed by the bibliographic application, remove the bibliographic application from contention for computing resources in an existing computer system, and lower total system cost, both initially and during continual operation of the system. This is the employment of specifically configured, highly modular, minicomputer-based hardware for the bibliographic task.

Some of the factors about large-scale computer systems to consider prior to determining whether to use a distributive computing approach for a bibliographic system are:

- processing priority in online and off-line modes available to the user;
- initial costs to acquire hardware such as terminals which are specific to the application and other central site hardware which might have to be added for the application;
- estimated continuing computer and system costs;
- availability of software facilitating development of the application;
- impact of centrally controlled hardware and system software changes which will affect the application and may necessitate reprogramming or additional maintenance programming; and
- system responsiveness in an online mode together with the ability to host or use devices specifically best for the application rather than one given device common and supported within the particular computer system and its system software.

With independence and user control of the total system resource also comes the responsibility to be able to maintain a functioning system. This does require appropriately knowledgeable systems specialists or the right kind of contract with an outside systems firm. In any such decision, there are trade-offs to be made. However, as most bibliographic applications do not require a large-scale computer system except to facilitate large array storage, enhance greater data transmission speeds to main memory and enable very large data base storage online, many individual agency or small to medium network systems can take advantage of a dedicated processing environment using minicomputer systems, either alone or linked to other computer systems, for a network with a system host processor. Each case is highly individualized but as minicomputers and their software take on many of the features of conventional computers needed by the bibliographic application they become increasingly attractive for such use.

MINICOMPUTER TRENDS

In the late 1960's and early 1970's minicomputers generally lacked the features which were needed for business data processing, text processing or largely character string manipulation problems. These mini's suffered from relatively slow cycle time, limited memory with lack of much expansion, lack of peripheral devices and their interfaces, unsatisfactory maintenance arrangements, small or inadequate instruction sets designed for numeric processing use, lack of system software and software development aids such as compiler level languages, debuggers, or diagnostics.

However, in the last five years many hardware and software advances have been made which make the minicomputer systems of today, exemplified in systems such as Digital Equipment Corporation's PDP-11 series minicomputers or comparable machines of other manufacturers, very attractive candidates for use in present and future bibliographic systems.

With limited word lengths, usually 16 bits (binary digits), only a limited number of words of main memory are directly addressable. Just how much main memory will be directly addressable is governed by the length of the instructions used with the given word length machine. Direct addressability simplifies programming but does require extra storage space and memory cycles to fetch the extra words used per multiple word instruction. However, with cheaper and faster storage it is a sound trade-off to use memory cycles in this manner.

Hardware advances are making standard many features previously not found on minicomputers but taken for granted on large-scale conventional processors. For example:

- techniques to address main memory beyond that directly addressable; indexing, indirect addressing of a single or multiple level nature, and virtual memory schemes;
- longer word lengths of 24 and 32 bits to permit faster data transmission rates and greater arithmetic precision, however at some significant cost differentials from most 16 bit machines;
- real time clocks or interval timers of programmable or non-programmable type;
- power failure protection and automatic restart capability;
- instructions in firmware or microprogrammable memory, either written by the vendor and implemented via a read-only memory (ROM) or by the user via a programmable read-only memory (PROM) and provision of a writable control store within memory instead of only a fixed control store to be accessed by the central processor;
- hardware implemented multiply and divide and floating point arithmetic;
- hardware byte manipulation;
- hardware stack architecture and provision of a number of general purpose registers which can act as accumulators, index registers, or as the program counter; and
- peripheral device hardware and software interfaces for a wide variety of equipment.

In a similar fashion software advances are continually being made, although generally in the field of computing software continues to lag behind hardware development. Some of these software areas are:

- various kinds of operating systems -- single and multiple user, supporting both sequential and direct access files, online and offline users and in multiprogramming or single tasking modes;
- improved system development aids -- assemblers (single and two pass), macro-assemblers, compilers, cross-compilers which permit program compilation and debugging in a large machine environment for transfer to the host minicomputer, linkers, loaders, debuggers, and text editors;
- improved software device interfaces supporting a wider variety of peripheral equipment;
- utilities such as report generators, screen formatters, sorts, merges, code translators, format translators, time and date routines, communication protocols, file handlers and file dumps;
- powerful data base management system software supporting host application languages such as COBOL, FORTRAN, RPG II, BASIC PLUS and various assemblers for specific machines;
- telecommunications and timesharing monitors; and
- extensive subroutine libraries available under various operating systems.

It is not our purpose here to describe these developments in detail in this paper but to let the reader know that software will play a much more important role in selecting a minicomputer than it did in the past.

CRITERIA FOR SYSTEM SELECTION

Once a decision has been made to use a distributive computing approach for a bibliographic system, then a host system must be chosen. Criteria for system selection are of two general categories. The first category comprises criteria related to the applications, their inter-relationships and their supporting environment within an agency. The second category can be derived from the first when it is understood as the second category relates to the actual hardware, software, system maintenance and operation factors.

Category 1 - Applications definition.

Bibliographic applications range in complexity from the integrated bibliographic management system required to acquire, catalog, index, retrieve, and circulate or distribute documents to specific product oriented systems such as those required for index production and publication, catalog card production or other mission directed activities common to a specific agency. For the most part today's and tomorrow's systems will be user oriented online systems wherein data entry, editing, file updating, and inquiry or retrieval are performed in real time via keyboard terminals and other specialized remote devices. Provisions for batch processing -- primarily to handle bulk data entry such as retrospective records or records from other sources used for system initiation as well as to handle various printed output such as short or long bibliographies, purchase orders, overdue and recall notices, claim notices, or any number of similar printed products -- are also an important capability of any online system even though the bulk of its work is done interactively. Thus, bibliographic systems will operate primarily in an online real-time mode.

The following criteria factors should be included in the application definition:

[1] Tasks to be performed by the system, their frequency and time requirements.
[2] Tasks requiring online, interactive support and those to be performed in batch mode.
[3] Data definitions for the records to be entered into the system, the tasks requiring specific data, and the data elements to be considered directly searchable and their inter-relationships.
[4] Estimates of the initial size of online files and their probable growth rates.
[5] Determination of the location and number of system users requiring online terminals and associated peripheral devices depending upon their application use and based on expected transaction volume, and/or location and nature of their relationship to the system.

The more that is known about the requirements that the system must meet, the easier it will become to determine the type of hardware/software environment best for that system. In general, minicomputer systems are initially configured as either stand-alone systems where all processing for the application takes place or as a preprocessor or front end minicomputer attached to another shared computer. As minicomputers have gained in features, gained in their ability to handle a wide range of large capacity mass storage devices, systems designers have found them more attractive for dedicated stand-alone use due to their relatively low cost, throughput and reliability. Moreover, in the larger sized systems configured in purchase price to sell for $125,000 to $250,000, powerful commercial data base management systems packages such as IDMS from Cullinane Corporation are available. Another data base software creation, IMAGE 3000 and its associated retrieval/inquiry subsystem QUERY 3000 are operational on Hewlett-Packard 3000 Series minicomputer systems.

Main factors which influence whether to use a minicomputer system in a dedicated, stand-alone fashion with, perhaps an extension of capability to communicate with other systems as needed for data and information access or sharing the processing in a front-end configuration are:

[1] Available main frame computers and any applicable software in an agency that could meet the storage and search requirements of the application in a less costly but adequately responsive manner for the user's requirements,

[2] Experience within the specific agency in segmenting applications in such a manner to use a minicomputer as either a pre-processor or a post-processor, freeing the host computer from code conversions, message formatting, editing, error handling, CRT screen formatting and report output which would free the host computer to seek and access data, perform updating, and system backup in the event of a pre-processor failure,

[3] The user's requirements for maintaining complete system control and integrity for security and time dependent operations, and

[4] The priorities, job scheduling, and accounting for billing main frame computing resources within an agency which can affect the operating cost and responsiveness level of the system in an adverse manner if the central system biases its priorities and its charge structure against input/output bound processing and large online resident files.

Category 2 - Hardware/software factors.

The computer system, its peripheral equipment and supporting software, chosen wisely, can aid the system development process, insure future flexibility for growth, and maximize dependable continuing operations at realistic costs. In a minicomputer system for bibliographic use, it is desirable to consider first the available central processors, their place in a family of processors having at least upward compatible instruction sets and with a choice of the necessary other devices available from that vendor. Looking at these peripheral devices from the minicomputer vendor will enable comparison to vendor products available from other companies marketing plug-compatible devices at more favorable price/performance. Main memory, mass storage, tape units of every type, printers and terminals are the common peripheral equipment devices which can be used in configuring a system to a particular price/performance level.

An increasingly important aspect to consider is the availability of vendor supplied software under which system development may occur and under which system operation will be supported. These include the use of languages and file management packages such as MUMPS (Massachusetts Utility Multiprogramming System) developed at Massachusetts General Hospital. This system is finding increasing use in bibliographic systems development with the work being done at the Lister Hill Center for Biomedical Communications of the U.S. National Library of Medicine. Another very satisfied MUMPS user is Washington University School of Medicine Library in St. Louis, Missouri which has chosen to create their network serials management system called PHILSOM III in this language. [2]

Even if the system is to be developed as a complete package under one vendor contract, it will be necessary to evaluate and monitor the key decisions affecting the creation of the system. A modular and flexible system of a generalized nature will enable future enhancement, fine tuning, and modification with less cost in programmer resources and less time toward completion of the change. A more specialized set of software will be easier to create initially but will tend to be less flexible to new requirements. With this introduction in mind let use now discuss hardware factors in greater detail.

Central Processor. In the United States, Europe and Japan there are currently 50-60 minicomputer manufacturers producing and installing various models of 8, 16, 24 and 32 bit word length minicomputers. Although nominally 8 and 16 bit processors are considered to be minicomputers, some of these larger models have power and features overlapping into the 24 and 32 bit machines. Therefore, in this authors opinion, all of these central processors should be looked at as highly capable generalized modular computers -- some of which decidedly lend themselves to character mode, text handling applications far better than others. Looking at these vendors, one can easily identify at least sixteen processors or families of processors which would be potential candidates for such applications as we have previously mentioned.

Most minicomputer central processors employ a multiple bus architecture as this is cheaper to build and simple in design. However, such machines require a separate input/output channel for each device. Another architecture, found in the Digital Equipment Corporation PDP-11 Series minicomputer, is the single bus type. The single bus architecture gives the user more flexibility to mix devices of various speeds together with the added advantage of direct input/output device communication with main memory, without central processor involvement. The only disadvantage of the single bus is its greater design complexity to accommodate varying data transmission speeds. Thus, few minicomputer manufacturers have employed this architecture.

Word lengeth for bibliographic use should be a multiple of 8 bit bytes. Longer word length processors offer greater precision in floating point calculation, larger directly addressable memory, and

increased data transfer rates. Generally, the longer word length processors have larger, more flexible instruction sets. These features and the incorporation of the number of accumulators or system of general registers, the input/output control scheme and other facilities of the central processor must be viewed as a whole.

Since the majority of minicomputers use parallel, binary processors with single address instructions, the number of accumulators can have a significant effect on flexibility and processing power. With multiple accumulators in this type of machine, instructions involving operands in the accumulator execute faster as they do not require retrieval of the operand from main storage before execution. Many machines now available use a series of general registers, one of which serves as the Program Counter and the others able to be used as accumulators or index registers.

To handle the problem of addressing more main memory than a single fixed word length of 8 or 16 bits permits, various schemes have been devised. One is indexing which is a form of address modification. Another scheme is to use single, double or triple word length instructions. To handle this problem, the number of index registers serves as an indicator of the flexibility and ultimate efficiency of the specific scheme chosen. Obviously, more main memory is required to store multiple word length instructions, but execution speed is usually improved. Also levels of indirect addressing, where the address part of an instruction specifies a storage location that contains another address rather than the operand itself, are employed. In indirect addressing each level of indirect addressing usually requires an additional storage cycle within the total machine cycles comprising total instruction execution time.

Input/output control features of the processor such as a direct memory access channel as a standard feature are important in a processor used for bibliographic work. In a multiple bus machine this feature permits direct transfer of data from a peripheral device controller to main memory without using the computer's main hardware registers. Once the input/output operation has been initiated by the program it can continue independently from further program control. In a minicomputer lacking this feature each word being transferred from a direct access storage device would have to pass through the processor's registers, interrupting internal processing operations and slowing the possible transfer rate of data.

In virtually every real-time, online interactive system, an effective program interrupt facility is a requirement. This is as true in bibliographic systems as in other online applications where simultaneous users are invoking many varied functions in unpredictable sequences. An interrupt is a condition where a temporary suspension of normal program execution occurs to permit dealing with whatever condition caused the interrupt. Interrupts are of two types - internal and external. Internal interrupts are caused by memory parity errors, illegal instructions, or power failures. External interrupts usually indicate that a peripheral device has issued a signal for servicing or has completed an input/output transfer. Contents of the Program Counter and the Program Status Word are temporarily stored followed by a transfer of control to a software routine that determines the cause of the interrupt and initiates an appropriate action. The number of external interrupt levels shows the number of different external devices that can be recognized and the power of the interrupt system on a particular minicomputer processor.

Main Memory. Although most current minicomputers use ferrite magnetic core technology, increasingly the newest machines are moving to metal oxide semiconductor (MOS) and bi-polar technologies. In some machines auxillary memory of either of these types is offered. In others the whole memory is of this type. Higher performance can be achieved through shorter memory cycle time with the the MOS and bi-polar products, but still at some increase in cost over ferrite core memory. Today's storage cycle times range from 850 nanoseconds to 3 microseconds for ferrite core memory, with MOS and bi-polar memory normally in the 200-600 nanosecond range.

In order to secure certain other features on some machines, a basic minimum amount of memory must be procured from the minicomputer vendor. For example, at one time the Digital Equipment Corp. PDP-11/40 required a minimum of 32K words memory before the memory management option could be secured. This enables addressing up to 124K words of memory in that machine. It is a type of virtual addressing scheme wherein the normal 16 bit direct byte address is no longer interpreted as a direct physical address but as a virtual address containing information to be used in constructing a new physical address of 18 bits incorporating the contents of the active page register. With plug compatible memory available for this computer which is 25%-40% cheaper than the vendor supplied memory, it could be very advantageous to order a minimum 16K memory module and then equip the computer later with plug-compatible memory of sufficient amount. Instead, a 32K minimum will need to be acquired to get this memory management option, necessary if over 32K words of memory are needed.

A key factor to consider is the total amount of initial memory required by the system to be developed. This will be determined by pinpointing the minimums required by vendor supplied software, other software such as data base management system products, estimates of the size of application software, the number of device buffers required or amount to be set aside for dynamic buffer allocation, and any reserved memory space which specific machines require. This latter aspect can be illustrated again by looking at the PDP-11 Series minicomputers. As these machines handle all input/output operations through a single bus called the UNIBUS, memory and device controllers are addressed alike through reserved addresses in memory rather than through a separate class of input/output instructions. These reserved addresses in the PDP-11 occupy the highest 4096 words of memory which can be addressed, thus reducing by that amount the total memory available for other purposes. However, this scheme does afford the advantage that each device controller can read and write to memory just like the central processor. This enhances central processor throughput and simplifies programming at the expense of some input/output overhead time.

Memory protection is now a standard feature of many minicomputers and generally offered as an extra cost option by other vendors. In the PDP-11 Series minicomputers, memory protection is a feature of this memory management option described above. In an online, multi-user system this feature is highly desirable to prevent programs from causing damage to each other through unauthorized writing in certain areas of main memory. Other machines use software means or a combination of hardware and software to accomplish this protection.

Parity checking is another feature which is more common as a standard offering in some manufacturers machines. However, it may be argued that reliability of modern electronic and semiconductor memories is so high that this is an unnecessary extra cost. Even so, many plug compatible memories offer parity checking at still lower cost than similar non-parity memory from the minicomputer manufacturer. In a larger system, where high transaction volumes or the handling of very critical data exist, parity is a desirable feature, worth any added cost. For other applications, and perhaps as technology continues to improve, it may not be considered a necessity.

Peripheral Devices. Very important to bibliographic systems is the configuration of peripheral devices that enable data input, output, storage, and system housekeeping or protection functions. In online bibliographic systems disk mass storage devices represent the most significant cost portion of system hardware. Each minicomputer manufacturer offers a line of peripheral equipment interfaced to his system, supported by his software and sometimes configured into packages designed for specific commercial applications.

In order to do sorting and merging three physical devices are needed. These could be a disk unit and two tapes or three disk units or another combination. However, in an online system sequential storage tape media are used primarily for disk backup copying, transaction recording for system backup, input of batch data or output of batch data for use in another computer system or for a later processing purpose.

As bibliographic records are lengthy, fairly large capacity devices are needed for most library applications, unless only rather current data is made available online. As the system grows, unless old data are removed from online storage, the disk storage will require an increase in its capacity. Depending upon the size and transaction load requirements of a system, removable disk pack drives from 20 million to 176 million bytes are available on a number of systems. Cartridge disk systems with small capacities of 2.4 million bytes are available in the smaller systems or may be used for system disk use. For program residency and swapping to main storage, fixed head disk systems can be added in sizes of 512,000 to 1.2 million bytes. In smaller systems, floppy disks can be used in place of cartridge disks or fixed head disks if speed is not a major factor. Floppy disks are now available with storage capacities greater than 300,000 bytes through improved densities and the recording of both top and bottom surfaces of the media. For system software, diagnostic loading, and similar use, floppy disks are becoming very popular due to their low cost and excellent reliability.

Industry compatible magnetic tape units with recording densities of 800 and 1600 bits per inch and reading speeds from 45 inches per second upward are generally available from the main minicomputer manufacturers. Cassette or cartridge tape drives using either the Phillips style cassettes or the 3M Company cartridges provide lower cost, lower capacity serial file mediums which many systems find quite adequate for transaction recording, diagnostic input, system loading, and program storage.

Printing capabilities in bibliographic systems range from short on-demand products such as user notices, bibliographies, book orders, circulation date due and fine notices to long printing of specialized bibliographies or catalog cards which require a very high quality, bottom feed printer mechanism. In a bibliographic system most such printing will occur at the user site through a remote printer device of serial character type and which may also incorporate a keyboard for use as an input device as well. Many units having an RS232C or current loop interface can be used depending upon print style requirements, cost, and character set requirements.

Most communication to a minicomputer system occurs through a terminal device having a keyboard. Since most interactive use does not require a line-by-line record of dialog, visual display terminals have become the main terminal devices for data entry, editing, and inquiry. In general, the display terminals offered by the minicomputer manufacturers are not very desirable for bibliographic system use as they do not have character sets beyond the normal upper/lower case 96 character ASCII set. Also, they rarely have the other features such as function keys, block mode transmission, protected format or other features of importance to a well designed bibliographic system. Thus, in choosing a minicomputer system it would be wise to reserve terminal choice to a separate procurement activity as generally different vendors will be involved. Where the system is to be a simple single application type, it may very well be possible to use these less sophisticated devices; for example, a simple circulation control system without full bibliographic information included in the implementation.

Data communications interfaces for systems involving more than one terminal device, whether all are physically proximate or remotely located from the building housing the computer need to be examined. Various alternatives are offered by the minicomputer manufacturers which enable hardware interfacing of terminals to computers and computers to other computers as well as the tying of computers together to operate in a multi-processor fashion in large minicomputer systems. Depending upon the nature of the application, the data communications provisions may assume a very important part of the overall evaluation.

In general, many other specialty devices such as optical wand equipment capable of reading bar-coded or zebra labels may be procured. Digitizers, plotters, color display devices, optical character reading devices, Hollerith card readers, paper tape readers and analog/digital data converters are some of the specialty devices available for interfacing to a minicomputer. However, the applicability of these devices is low except for the use of the optical wand reader in circulation recording systems or large processing center control of the status of items proceeding through the center for cataloging, physical preparation, etc. Again, if these kinds of devices are to play a part in the system, their procurement would be done best by a separate specification and bid process.

Software and Instruction Sets. Today virtually every major minicomputer vendor offers a wide variety of software. Operating systems, assemblers, macro-assemblers, compilers, communications control, file management, data base management systems, linkers, loaders, text editors, utilities such as file dumps, disk compression, and diagnostic routines represent the range of system support and development software available. Unfortunately, few applications programs for bibliographic systems are available but a number of single application oriented systems are available for circulation control from their developers. However, soon some library systems developed under the MUMPS data base and language system on Digital

Equipment Corporation PDP-11/34 and larger systems for library and information retrieval system use. A data base management system called PATHFINDER, written in the MUMPS language, currently operating under MUMPS, but planned to eventually operate under other PDP-11 operating systems, has been developed at the U. S. Drug Enforcement Administration.[3] This system is being used to support a very clever searching scheme involving multiple yet related families of information. This software could be used to build an integrated online library management and retrieval system. This agency is planning to enhance this package to operate under one or more of the most powerful operating systems available for the PDP-11 Series larger machines. Another system could well emanate from other work going on in the United States, Great Britain, or Australia.

Except for single purpose systems such as circulation control, library systems of an integrated online nature, with multiple terminals, large files, and diverse application tasks which are largely input/output dependent require systems configured on the medium to large scale end of the minicomputer ranges. If data base management system packages, such as DBMS-11, on the PDP-11 are used for the heart of such systems, these require the larger machines as higher level languages such as COBOL are used for applications brought up under the data base management system. A rather sophisticated operating system and large memory are required in such a system. However, the number of users to be accommodated, the flexibility to accommodate very large and complex data structures make this an attractive system for these larger library bibliographic systems.

In just the last several years vendor software has now reached the point that it can support bibliographic system development so that the user does not have to start with a bare machine and develop his own operating system, utilities, software device drivers, file managers, disk access and allocation routines, and other enabling programs. Thus, vendor software will be of increasing importance in making a choice of system depending upon whether the system is to be a single or multiple user type, or serves a single task or is to be highly integrated.

The instruction set of the minicomputer in some machines is implemented through microprogrammed sequences stored in a read-only memory (ROM) in addition to the conventional hardwired logic portions. Some machines offer user microprogramming ability via a programmable ROM or PROM memory. In the ROM units these are non-alterable by the user and are created by the vendor. Microprogrammability can greatly increase the flexibility of a minicomputer, but here, again some trade-offs of reduced speed and increased price will be encountered.

Most minicomputers are weak in data test type instructions as they have no COMPARE instruction. This is one area where some machines could be enhanced by proper microprogramming. Instructions such as TRANSLATE are also not found on minicomputers but macros in the programming can be used to implement such instruction capability from a macro library.

For bibliographic applications byte and bit manipulation are necessary capabilities. If this is effected by hardware features rather than through software, processing speed will be increased. Immediate or literal instructions are available in some minicomputers which save storage and execution time. An immediate instruction uses its address field to hold the operand itself rather than the address of the operand, thus saving the time to fetch the operand as well as its storage space.

Hardware multiply and divide capabilities are useful in some of the business and statistical portions of bibliographic systems, although programmed subroutines can be substituted where this capability is not a hardware feature or option. Normally hardware floating point instructions are not available in the lower priced minicomputers and only in the larger minicomputers is this capability offered, sometimes as a standard feature but more often as a separately priced option. Hewlett-Packard HP 3000 Series computers offer hardware floating point as standard while the Digital Equipment Corporation PDP-11/70 offers it as an extra cost option priced at $5,600.

A real time clock and/or interval timer is quite necessary to any online system so that programs can determine time of day or measure intervals to trigger interrupt signals. Power failure protection and automatic restart is also vital in this type of application.

OTHER EVALUATIVE CRITERIA

In choosing a minicomputer, evaluation of the vendor's ability to provide service and support is usually critical except in rare cases of certain large agencies having internal hardware servicing support. Without a good relationship between user and vendor, what may be a technically superior hardware system will be unsuccessful in operation if it cannot be readily serviced. Thus, in particular locations throughout the world, systems to perform the same tasks may, of necessity, come from different manufacturers. This may be true even though a system may lack certain technical merits as often a best compromise choice has to be made.

Delivery schedules of vendors may also influence choice, particularly in heavy demand periods for specific systems. However, sometimes a third party procurement can be arranged through a lessor or system broker who essentially is purchasing machines in demand and then selling his place in the delivery queue. The vendor's ability to supply the major portion of the hardware and appropriate software according to customer specifications or policies will particularly influence choice in the smaller systems.

Multiple vendor systems do have problems, but in a bibliographic application multiple vendors are inevitable if a well configured and cost effective system is to result. At least a minicomputer vendor, a terminal vendor, and some special device vendor for optical wand readers will normally be found. However, in a well managed and directed enterprise, improved performance at some significant cost savings can result from a careful selection of plug compatible devices, depending upon the customer and his pricing qualifications with the individual vendor -- such as educational and quantity discounts or pre-paid service discounts.

CONCLUSION

From our discussion here, it can be inferred that selection of a minicomputer system for bibliographic applications involves definition, organization, decision trade-offs, cost considerations, varying procurement procedures, and much good common sense judgement based on technical experience. Whether the system is large or small, the same organized approach and considerations should basically apply. In a paper of this length it is only possible to cover the most essential factors of hardware/software selection which apply to bibliographic systems. Certainly, as one proceeds toward system selection other considerations may also have to be incorporated which have not been touched on here. Some of these could relate to specific contract provisions of purchasing or leasing or obtaining a completely packaged minicomputer system composed of hardware, application programs and supporting systems software.

REFERENCES

1. Grosch, Audrey N. "Minicomputer Characteristics, Economics and Selection for an Integrated Library Management System." In: Clinic on Library Applications of Data Processing, April 28-May 1, 1974. Proceedings: Applications of Minicomputers to Library and Related Problems. Edited by F. Wilfrid Lancaster. Urbana: University of Illinois, Graduate School of Library Science; c1974. LCN 65-1841 ISBN 0-87845-041-6 ISSN 0069-4789. p. 158-169.

2. Johnson, Millard F., Jr. "A Design for a Minicomputer-based Serials Control Network." In: Special Libraries Association. Annual Conference, 66th, Chicago, IL, 1975 June 8-12, Contributed Papers Sessions: Systems and Networks: A Synergistic Imperative. 9p. Paper No. 13 of 26 papers on microfiche, 24X reduction. Available from: The Author, Washington University School of Medicine Library, 4580 Scott Avenue, St. Louis, MO 63118 U.S.A.

3. Levine, Emil H.; Steve E. Rodgers; Stanton E. Mintz. "PATHFINDER I - A Minicomputer DBMS - Low Cost High - Powered Computing." A Poster Session at the 40th Annual Meeting, Chicago, IL of the American Society for Information Science, September 26-October 1, 1977. Abstracted in Proceedings of the ASIS Annual Meeting, Volume 14. New York, Knowledge Industry Publications Inc. for ASIS, 1977. ISSN 0044-7870 ISBN 0-914236-12-1. p.46-47.

CIRCULATION CONTROL

by
Christopher J. Aslin
Sub-Librarian
University of East Anglia Library
Norwich
Norfolk NR4 7TJ
U.K.

SUMMARY

It is possibly true to say that libraries contemplating using computers to help with housekeeping and management will often consider circulation control as one of the first systems for conversion. This is indicated by the relatively long history of such systems.

Though the use of minicomputers in libraries is a more recent innovation, their use in circulation control is now well established. However, their mode of application is seen to be variable and dependent on the local circumstances and priorities which a library places on various aspects of its system management. Examples of minicomputer circulation control systems are described to illustrate how the computer has been utilized in quite different ways to cope with the problems of differing library requirements.

INTRODUCTION

The use of computers in circulation control is now well established although in fact it is not much more than a decade since computers were being considered as suitable aids to library housekeeping routines for the first time. For example, publications of the mid-sixties such as that by Harvey (1) were referring to circulation control systems which relied largely on the physical sorting of transaction cards as the central process. Progress through the 1960's into the 1970's saw a rapid development and implementation in this field. The three papers by Wilson (2), Lingenberg (3) and McCann et al (4) are interesting in that over the few years they represented it is possible to see the rapid increase in sophistication of systems being used. The paper by McCann et al also refers to one of the earliest instances of minicomputers being associated with circulation control. This was at Bucknell University Library, where it was used as a back-up device in case of failure of the mainframe machine.

The rapid development of on-line systems in the early 1970's together with the even faster developments taking pace in minicomputer production led inevitably to the minicomputer-based circulation control system. The conference proceedings edited by Lancaster (5) and the papers from the LASIE Workshop on on-line circulation control systems (6) serve to give some indication of the state of the art in the U.S. and Australia at that time, with Foil & Carter (7) providing more recent information for the U.S.

In the U.K. the same process has been taking place as indicated by publications by Young (8), Gallivan (9), Hudson (10), Partridge (11), and more recently, by Aslin (12), Green (13), Pickles (14) and Wilson (15).

There is obviously a great deal of information either already available or being produced at this time. I thought it worthwhile, therefore, to concentrate on the system which I have been responsible for developing at the University of East Anglia (UEA) and comparing aspects of it with other existing systems, one a large public library, the other a co-operative academic library system. The UEA system illustrates the high degree of control that minicomputers place in the hands of the issue desk staff, and it is also possibly one of the first systems developed which has the potential of being completely independent of any mainframe computer. Until now this system has not been described in any great detail in print.

THE UNIVERSITY OF EAST ANGLIA CIRCULATION CONTROL SYSTEM

Background

The University is one of the 'new universities' and took in its first students in 1963. There are almost 3,500 undergraduates in the year 1977/78 and the Library has some 5,500 borrowers on its files. Book issues were about 238,000 for 1976/77 not including the Restricted (short) Loan collection.

Prior to the introduction of the present system the Library operated an off-line batch processing book circulation system based on that developed by the University of Southampton (16). The data input which this required involved the use of an 80-column punched card in each book and a punched plastic badge to identify the borrower. Issue desk transactions were accomplished by using the cards and badges as necessary to produce paper tape for computer processing via Friden Collectadata terminals. As quite a large amount of effort, and therefore capital investment, had been necessary to produce

machine-readable book records, when it became necessary to replace the ageing terminals, it was decided to try to retain the punched card form of input. Experience with the old system had also shown that there was an advantage in inputting alphanumeric data at the time of transaction. This helped clear up problems which could arise due to imperfect data collection and transmission - a problem frequently encountered with the electromechanical Collectadata machines. It was also felt essential that information being supplied by the system either to Library staff or borrowers should contain as much detail as possible.

A survey of possible computer-based circulation systems then in existence or under development led us to the conclusion that the Singer System Ten computer (now ICL System Ten) with its associated Model 100 Job Input Station (JIS) terminals held the greatest potential for providing a system compatible with our methods of data collection and would allow a large increase in control with only a relatively modest commitment to software development. The System Ten is designed basically as a transaction processing machine and was therefore well-suited to the procedures which are characteristic of the day-to-day running of an issue desk. The capital investment on machinery seemed large, (about £30,000, equivalent to over $60,000 at that time) but it was felt cost-effective when all other considerations were taken into account.

The McKeldin Library at the University of Maryland, U.S.A., had been using this equipment for their circulation control (17). But there it was being used to collect daily transactions which were then processed overnight on the University mainframe machine. This kind of hybrid system using a mini- and mainframe computer had also been developed at the University of Lancaster in England (9). It seemed to us, however, that as we were a smaller university than Maryland the System Ten computer with a single 10 million character disc had the capacity for holding all our files on-line and not just the daily transactions. This was borne out by some studies on the old system, some results of which were reported in 1976 (12).

At about this time the Singer company also issued new software, Disc Management Facility Mark II (DMF II), designed to run on their up-graded Model 21 processor. This has the facility for direct accessing of files in the random mode, thus allowing potentially rapid access without the re-indexing problems which can occur with indexed files. There are problems associated with random access also, but it was felt that the advantages outweighed the disadvantages.

Experience with our earlier off-line computer-based system had demonstrated the advantage of co-operating with another library in using a developed system and making only minor alterations to suit our own environment. Although one may not end up with one's own expectation of a perfect system and will have to change some existing procedures and designs to accommodate the 'foreign' system, there can be a large gain in other respects, such as speed of installation and, one hopes, tested and reliable programs. Obviously, if there is no computing expertise available to the Library then there is little choice in this matter. In our case as software supplied with the machine covered disc accessing, file handling, terminal control and communications, then all that was required was to develop the software used to process the data being entered into the system, to store it, and to allow access and removal of information as necessary. In the event this took about six months.

A small advisory working group was set up with the purpose of working out the detailed requirements of the system. On occasions larger meetings of the Library staff were reported to with the object of keeping people in touch with developments.

As the machinery had been delivered some months before the scheduled 'go live' date we had the advantage of developing the software in situ. Also, the advisory group and other members of the Library staff most involved could see how things were progressing and if necessary propose changes. Having machinery on site during the development phase was extremely valuable as it allowed a flexible approach to the system design and allowed staff to become familiar with the terminals and their responses. With this kind of development there is of course a cost to be considered. Expensive machinery is not, seemingly, doing productive library housekeeping, and one must make allowances for the possibility of a slightly longer development time than might have been the case if developments had taken place elsewhere with the use of a detailed specification of output and input requirements.

The Hardware and Software

The System Ten configuration purchased for the circulation control system consists of a DCS (Data Collection System) comprising:

1 Model 21 Processor with 40K characters of memory
1 Model 40 Disc Drive with a 10 million character disc
1 Model 45 Magnetic tape unit
1 Model 70 typewriter Workstation
3 Model 100 JIS terminals

Subsequently we acquired a serial printer to facilitate the printing of large amounts of data.

The software supplied with the machinery includes such things as a Monitor, clock program, terminal driver, together with utilities such as an Assembler, Editor and File Maintenance program.

The processor allows hardware partitioning of memory by placing loop connectors in particular positions so that the memory can be divided up into blocks or partitions. Each partition is of a size to accommodate a resident program. Slow peripheral devices are connected via a hardware input/output control (IOC) to the partition containing the relevant program. Adjoining partitions and their associated peripherals are completely independent and can be working on other tasks. Part of the memory is called Common Memory and is accessible by all partitions. This contains all the standard file access routines compiled into a program called LIOCS (Logical Input Output Control System). These routines are used for accessing the fast peripheral devices, the disc and tape, via the hardware File Access Channel (FAC). Any program too large to reside in a partition, i.e. over 10K characters in our machine, can either overflow into the common area or can be overlayed from disc. Automatic switching from one partition to another provides slices of processing time for each program. Consequently, multi-programming can take place without the requirement of a software executive or operating system and this saves core space. A guide to the layout of computer memory is shown in Fig. 1.

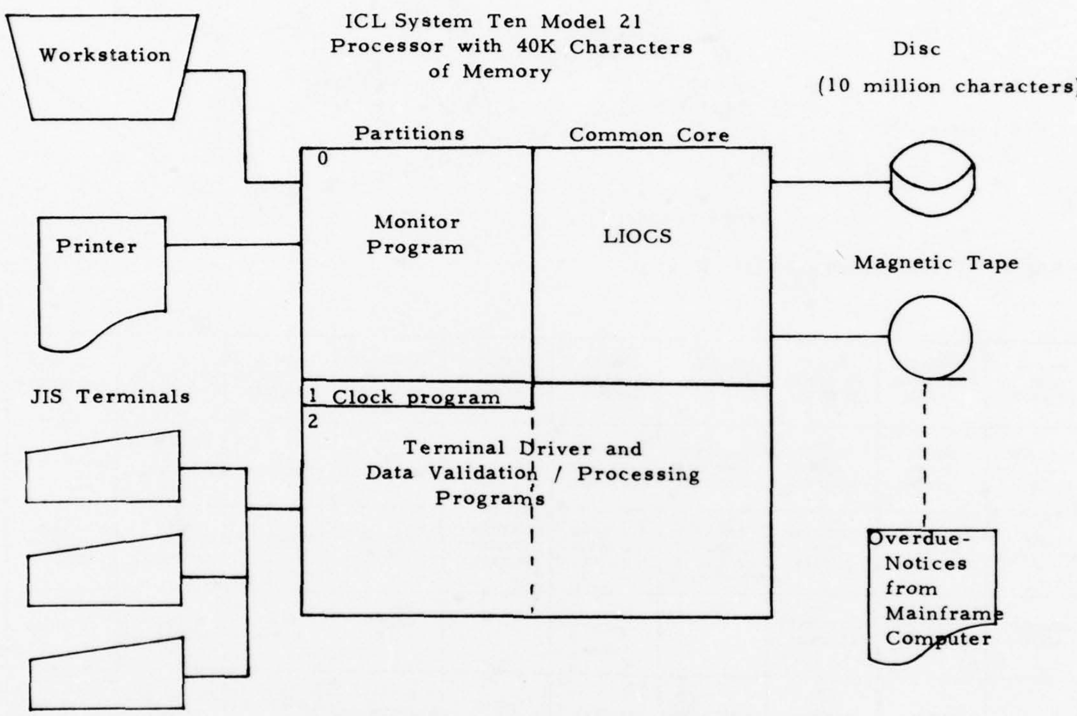


Fig. 1. UEA Library System Outline and Core Memory Layout

The ten million character disc is arranged in 100,000 sectors 100 characters. This does not mean that records always have to be 100 characters in length, but this factor has to be kept in mind when the size of records and files and the efficiency of disc space utilization are taken into consideration. Records are grouped in files, and files in pools. All files in a particular pool would normally be of a similar kind, e.g. object files or source files, but when relative and random direct access data files are used then only one file is allowed in each pool.

The magnetic tape unit is used for data security reasons in case of disc failure. All daily transactions are written to tape as well as disc. Magnetic tape is also used for communicating with the University ICL 1903T mainframe machine.

The workstation is the main control device and provides a means of inputting data and printing system messages.

The JIS data collection units (Fig. 2.) are the successors to the Collectadata machines mentioned above. They are sophisticated intelligent terminals capable of imputting alphanumeric data read from 80-column cards, punched plastic badges and a numeric keyboard. A windowed display screen is provided with a photographically-produced mask containing up to 50 fixed messages (Fig. 3). In addition at the top of the display area lighted numerals show either the time of day or the variable numeric data being keyed in.

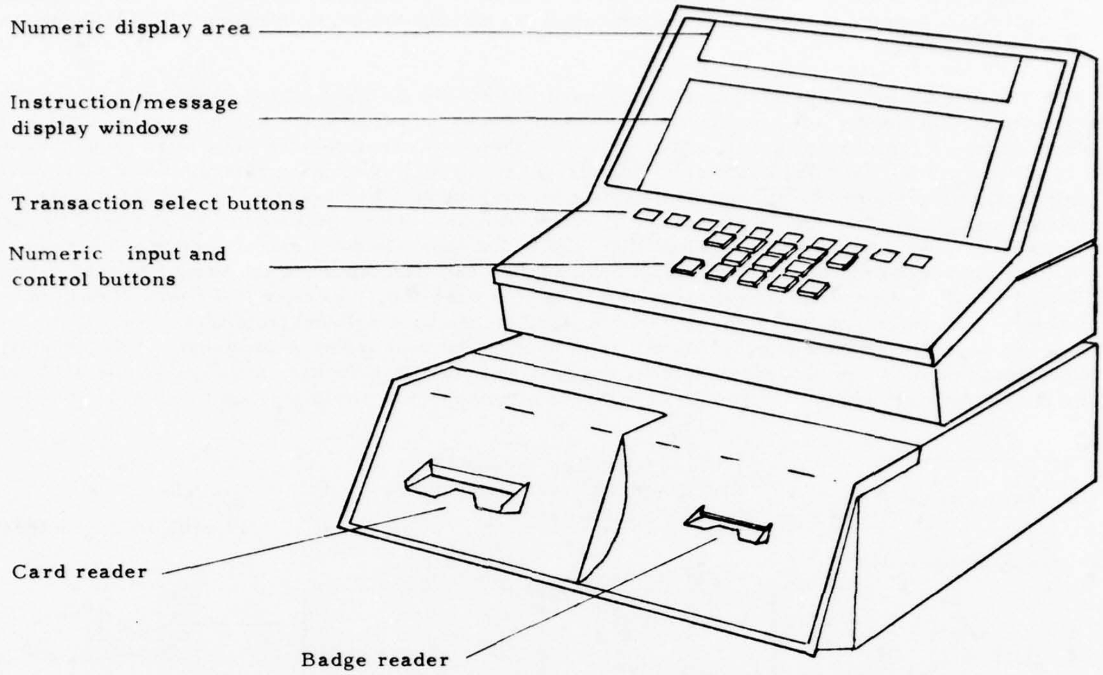


Fig. 2. Issue Desk Terminal

KEY IN CODE NUMBER	INSERT BORROWER BADGE	KEY IN BORROWER NUMBER	KEY IN RETURN DATE	INSERT RENEWAL BADGE	INSERT AUTHORITY BADGE	RESERVED BY ANOTHER BORROWER	KEY IN RESERVER'S NUMBER	KEY IN TRANSACTION NUMBER	SELECT TRANSACTION
ALREADY ON LOAN	INSERT BOOK CARD	KEY IN BOOK NUMBER	BOOK NUMBER INCORRECT	BORROWER CATEGORY INCORRECT	INSERT PUNCHED CARD	NOT RESERVABLE	NOT RESERVED	BOOK NOT ON CIRCULATION FILE	TRANSMITTING
CODE INCORRECT	BORROWING LIMIT EXCEEDED	BORROWER NOT ON FILE	RETAIN BOOK FOR ISSUE DESK SUPERVISOR	RESERVED (RETAIN BOOK)	BORRWER NUMBER INCORRECT	BADGE ERROR	PROHIBITED BADGE	KEY IN BORROWER CATEGORY	ACTION &/OR ERROR
MISSING BOOK	EMERGENCY ISSUE (NO BADGE)	EMERGENCY RENEWAL		TRAP BOOK		BADGE TEST	TRAP BADGE	CANCEL RESERVATION	RE-ENTER
	ISSUE	RETURN	EMERGENCY RETURN	RENEWAL	ISSUE (3 DAY LOAN)	RESERVATION	LOCATION ENQUIRY	EMERGENCY ISSUE (GREEN CARD)	SPECIAL TRANSACTION

Fig. 3. Display Screen Messages

The fixed messages are generated by lights behind the mask, and they are of two kinds: either instructions to the operator as controlled by the terminal resident programs, or messages produced as a result of the core resident program working on the data input.

The system can deal with up to 90 different transactions although at UEA we use only 27. Of these, the nine most common are stored in the terminal memory thus allowing faster access, and individual transactions are selected by pressing one of the buttons below the bottom row of illuminated legends. Under normal idling conditions these are all lit. Any less common transactions are selected by pressing the special transaction button and keying in the number required so that it can be loaded from the computer main memory. For example, to issue a book, assuming that both borrower badge and book card are present and that it is not a short loan book, the issue desk assistant presses the button below the window marked 'Issue'. The terminal program switches off all other lights except 'Issue' and lights up the window marked 'Insert Borrower Badge'. The assistant then inserts the badge and the terminal checks that it is inserted correctly and has enough holes punched in it. If all is well it gives a faint audible signal and lights up the window with the message 'Insert Book Card'. The terminal reads the card and checks that it has also been inserted correctly and gives an audible signal that all is well. If at any time the card or badge had been inserted incorrectly the 'Re-Enter' light is flashed and a repeated warning sound is given. Pressing the 'Enter'button transmits the data accumulated by the terminal for further processing. The 'Transmitting' window remains lit while this takes place. If all validity checks and file updates have been successful the terminal relights the 'Issue' and 'Insert Borrower

Badge' windows to allow the next issue transaction. If any data or attempt at file update was found invalid then the 'Error' window flashes, the relevant error message window is lit and the repeated warning sound given. If the circumstances of the error are such that more explanation is necessary then details are printed automatically on the workstation. The error flashing light and sound can only be stopped by pressing the 'Cancel' button. In an error-free condition this also serves to place the transmitter in the neutral or idling mode to allow selection of other transactions. The numeric keyboard is used for enquiry transactions or in cases where the book card or borrower badge is absent. For example, selecting the Location Enquiry transaction lights the window 'Key in Book Number' and eight zeros are displayed on the numeric display panel. As each digit of the book number is entered it appears on this panel and allows the assistant to check the accuracy. Pressing 'Re-Enter' will return the panel to zero again allowing repetition of data input. A list of the transactions is shown in Fig. 4.

Transaction	Input		
	Badge	Card	Keyboard
Issue Transactions			
NORMAL - Standard Loan	x	x	
" - Three Day Loan	x	x	
" - Variable Loan Period		x	x
EMERGENCY - No Card - Standard Loan	x		x
" - " - Three Day Loan	x		x
" - No Badge - Standard Loan		x	x
" - " - Three Day Loan		x	x
" - No Card or Badge - Standard Loan			x
" - " - Three Day Loan			x
Return Transactions			
NORMAL		x	
EMERGENCY - No Card			x
Loan Renewal Transactions			
NORMAL - Standard Loan	*	x	
" - Three Day Loan	*	x	
EMERGENCY - No Card - Standard Loan	*		x
" - " - Three Day Loan	*		x
Reservation Transactions			
NORMAL	x		x
EMERGENCY - No Badge			x
SPECIAL CARD INPUT	*		
Other Transactions			
MISSING BOOK ISSUE	*	*	
CANCEL RESERVATION	*		x
TRAP BOOK	*		x
TRAP BORROWER	*		x
BADGE TEST	x		
BOOK LOCATION ENQUIRY			x
BORROWER FILE ENQUIRY			x
RESERVATION FILE ENQUIRY			x
AUTHOR/TITLE INFILL		*	

x - Type of input used
* - Special Badge or Card

Fig. 4. Circulation Control Transactions used at UEA

Validation

Besides the checks carried out by the JIS terminals the system programs check automatically on such things as:

Modulus-11 check on Borrower number
Borrower on file
Borrower badge not out of date
Borrower not trapped for any reason
Borrower is not about to borrow over the limit
Book number Modulus-10 check
Book not already on file at time of issue
Any book traps
Whether a reservation is possible
Whether a loan renewal is possible.

Files

It had been decided that the system should be completely on-line with all data accessible for immediate updating after appropriate validation checks, and that answers to enquiries should be as immediate and as comprehensive as possible. With this in mind the speed of access became important. The number of file accesses and data transfers in a circulation system is potentially very high, and depends on parameters such as the length of loan period and the maximum number of books allowed on loan to a borrower. The fastest possible access would have been the relative access or self-addressing method where each record has an absolute fixed address relative to the beginning of the file, and its identifier, the book number for example, would have been directly related to its disc address. This method would have required a large commitment in disc space, in effect, one record for each borrowable book, whether or not it was on loan, and in our case quite out of the question. The slowest method would have been serial access and obviously also out of the question for a real time system with large files.

The choice available to us lay between indexed files and random direct access. These give approximately equivalent access times when first set up, but where a high turnover of data occurs the indexed files become inefficient fairly quickly and system response times increase. The indexes have to be recreated quite frequently and this is a time-consuming task. With the random direct access method each record has a key which is subjected to an arithmetic manipulation by a hashing or address generation algorythm. The result of this is to produce an absolute disc address relative to the beginning of a file, and whenever the disc is accessed the hashing algorythm is used to find the relative address. This means that, theoretically, all records have an equal chance of being found on the first try. Files of this kind, however, have their problems. There are about 300,000 borrowable books at UEA, but as previous studies had indicated that we were unlikely to have more than 25,000 books on loan at any one time the circulation file size had been set at 30,000 records. The key used, i.e. the book number, is eight digits in length. Therefore, some of those 300,000 eight-digit keys in trying to produce an address between 0 and 29,999 are going to produce identical addresses for different book numbers. This phenomenon, sometimes known as overflow, gives rise to chaining of records. If an incoming record finds its first choice position already occupied then it is placed in the next available space. With time the problem becomes worse and lengthy chains can be built up causing a degradation of response time. For optimal working the file should be about 80% full. Below this there is a waste of disc space with very little improvement in response time, whilst above, a rapid increase in response time takes place. Deleted records remain on the file in order to retain the integrity of the chains and allow overflowed records to be found. They are also sometimes used in providing information about the past status of a book. But eventually they have to be removed. This is accomplished by a set of programs designed for that task, the File Clean-up Sub-System programs. The circulation file is cleaned up every two weeks in term time, the other files less often.

Because random direct access files are suitable only for fast retrieval of information via one key, and as it was a requirement that book or borrower information be supplied on demand it became necessary to maintain separate files.

The main files maintained by the system are the Circulation File, the Borrower File and the Reservation File. These are all random direct access files. In addition, the daily transactions are stored in a variable sequential file. As all files are on line there is no need for a separate trapping file for reserved books, delinquent borrowers, etc. All records are capable of being trapped by setting flags. This is done by individual transactions, such as 'Trap Badge' or by the program logic, for example, on the return of a reserved book. A study we made of the usage of the old circulation system applied to a prediction of probable usage of the new system indicated the file sizes that would be needed. In two cases this was overestimated.

The Circulation File

This is a file of displaced items. It includes books on loan to individual borrowers, missing books, books at binding, books reported lost, on inter-library loan, books which are being repaired and those which are in the short loan collection. It should theoretically contain a record for any item not on the Library shelves, except of course, books which are being read in the Library, or waiting to be reshelved. The Circulation File record is a simple one with a fixed length of 100 characters of which at present 94 are used (Fig. 5.). It occupies 30,000 sectors of disc space.

The Borrower File

This is a file of registered borrowers and the books they have on loan. It contains records of two different kinds (Fig. 5.), one, the borrower details, the other the book numbers of borrowed books.

The main problem with designing this file was that allowance had to be made for, on the one hand, a borrower with no books on loan, and on the other, special Library 'borrowers' such as the short loan collection which may have 3,000 or more items on loan. To provide every potential borrower with enough file room to accommodate the maximum number possible would have required a great deal of

disc space. The solution adopted was to add a digit to the borrower number thus providing an eight-digit key. Every borrower must be on file for the system to accept any transaction involving him, and therefore his borrower details are stored in the record whose key includes a zero. As soon as the first book is charged out the system allocates an additional record of 100 characters giving the borrower number an extra digit '1' to form the key. Up to ten book numbers can be stored in that record. When the borrower takes out his eleventh book another record is created with a '2' and so on up to nine additional records, thus allowing up to ninety items on loan to any one borrower badge. 'Borrowers' such as Binding and Restricted Loans are catered for simply by providing them with a block of badges whose numbers have a common four-digit root. This provides a theoretical limit of 9000 books to any one borrower.

As soon as enough books are discharged to empty a record then it can be allocated to the free file space for use later by the same or a different borrower, as the case may be.

The Borrower File was originally allocated 15,000 sectors of disc space but this kind of dynamic file processing has meant that even when the system is on maximum loading only about 10,000 sectors are used. Applying the 80% rule it was therefore possible to consider reducing the file size and to allocate the saved disc space to other tasks.

The Reservation File

This is a file of reserved books with the reservers' borrower numbers (Fig. 5.). A maximum of three reservations per book is allowed as experience has shown that a longer queue was pointless since, by the time the fourth potential borrower was notified that the book was available it was usually too late. The file was originally allocated 4,000 sectors of disc space, but one surprising and gratifying fact to emerge was that the new system improved the turnover in reservations so much that it is now rare for more than 400 books to be reserved at any one time. The file size was therefore cut by a half and could be reduced even further if disc space becomes a problem.

CIRCULATION FILE RECORD

Book Number	Author / Title	Classmark	Return Date	Borrower Category and Number	Reserve & Exception Flags	Overdue & Recall Counts

BORROWER FILE RECORD (Sector zero)

Borrower Number	Sector Number	Borrower Category	Borrower Name	Title	Address Code	Borrower Stop Code	Badge Expiry Date	Lost Badge Link No.

BORROWER FILE RECORD (Sectors 1 - 9)

Borrower Number	Sector Number	Sector Constant	Stored Book Numbers

RESERVATION FILE RECORD

Book Number	Reservation Count	Reservers	Reserve Shelf Indicator

Fig. 5. Record Fields (Not to scale)

Day-to-Day Running

I have already referred to the JIS terminals and made passing reference to the Workstation. This latter machine, rather than a Visual Display Unit, is used as a controlling device, due to the need for hard copy messages. It has a normal typewriter keyboard with additional control keys. Illuminated fixed messages on a screen indicate the status of the machine, 'On-Line' or 'Local' for example.

First thing in the morning it is used to start up the system and open the files. It is then continually on-line under the control of the 'Monitor' program. Among other things the task of the Monitor program is to route messages from program modules which are processing the data input via the JIS terminals. Normal error-free transactions ellicit no response from the system other than to reset the JIS terminals to the starting position. But file enquiries will either light up the relevant window if a record is not found or print out details on the Workstation if it is present on disc. Should any transaction result in an error situation then, as well as a notification on the terminals, a suitable message is printed automatically on the Workstation, for example, the detection of a missing book or the re-issue of a book which has not been discharged. The return of a reserved item would also result in a printed message. This would contain information on the name and address of the reserver, and shortened book details (Fig. 6.).

The Workstation is used for all other business associated with the circulation system. Borrower file maintenance is carried out with a specially-written program. This allows the placing of new borrowers on the file, the alteration of existing records, enquiries, including the listing of books on loan to a particular borrower and the deletion of records of borrowers who are no longer with us. It is a conversational program giving the operator guidance at each stage. This program is used on demand and is loaded by removing the Monitor program first. Should a JIS transaction require the printing of a message the system halts until the Monitor is brought back into use (Fig. 6.).

```
05-40PM
75017211;LEES,D./IMPAIRMENT,DISABILITY & HAND     GM2410  77333  23000818   19 2
05-41PM
75017211  RESERVED BY * 7500629
05-41PM
7500629 2 MISS INGLISH H V                           SOC 0  06  780600
05-41PM
INGLISH H V                  MISS SOC   75017211;LEES,D./IMP         RS AVAIL
05-41PM
MONO
I)MONITOR ROLL-OUT COMPLETE
A)17.42.03: ENTER PROGRAM NAME.
YUIOP

TYPE OF ENTRY (A,D,E OR N)?  E
 TYPE IN BORROWER NO *   4000137
        400013701ASLIN C J                            DR  LIB  0 03 800600

LIST REQ.? (Y/N)  Y
   LIST OF BOOKS ON LOAN TO BORROWER NO 4000137

      76132661  STEADMAN,P./ENERGY,ENVIRONMENT & BUILDING.75          TJ163.2     77187   9   4
      13069007  KIMBER,R.T./AUTOMATION IN LIBRARIES.68                Z455        78081   9
      75087456  PICARD,M.D./GRIT & CLAY.75                            UE35        77348 1 9 1

 TOTAL  03
ANY MORE? (Y/N)   N
FINISHED
A)17.42.51: ENTER PROGRAM NAME.
START
  LIBRARY MONITOR IN
05-43PM
75017211  RESERVED BY * 7500629                   RESERVE
05-43PM
75017211;LEES,D./IMPAIRMENT,DISABILITY & HAND     GM2410  77343  RESERVE    55
05-45PM
76077315;MOTT,N.F./ELEMENTARY QUANTUM MECHANICS.72          7612036  *SEE NOTE 1*
```

Fig. 6. Examples of Workstation Messages and the Use of the Borrower File Maintenance Program (YUIOP)

At the end of the day the Workstation is used to close down the system, load the program used for analysing the day's statistics, and if necessary, run the program for making security copies of the files.

The speed of the system depends on several factors. They include the fundamental machine speed, together with the amount of data being transferred, the number of file accesses, the state of the files and the number of terminals in use. The fastest responses range from just under three seconds to complete an Issue transaction to just over one second for a Location enquiry. This is not so fast as some systems but is quite acceptable in our case when one considers that in that time it has carried out all the validation checks and filed the information in such a way that all files are completely up-to-date and available for any enquiry other than one requiring cumulated statistical information.

In addition to the main circulation control programs, which were designed to operate on-line in real time, we have developed a collection of batch processing programs for various purposes. The File Clean-up Sub-System has already been mentioned. There are also programs for checking that the files are in good health and are not being corrupted by any system malfunction. Programs for translating System Ten USASC II code into 1900 ISO code and back again have been written so that we can transfer data from one machine to another, together with sorting and printing routines for statistics production.

Now that we have acquired a printer the UEA Circulation System is capable of standing alone. At the moment the University mainframe computer is used for printing overdue notices as this still requires the sorting of large files. When time allows it will be possible to redesign our procedures so that, if required, overdue notices could be generated in the Library. At the moment it is more convenient to use the large machine.

Some Comparisons with other Systems

For reasons I have already outlined I have concentrated on one particular minicomputer system but it is interesting to look briefly at two other systems which use minis in circulation control. Both are from the U.K., one is the London Borough of Havering Public Libraries and the other is SWALCAP (South West Academic Libraries Co-operative Automation Project). They have been described in other publications (10, 11, 13).

Both systems are large in concept, Havering serving ten public libraries and SWALCAP three academic libraries at Bristol, Cardiff and Exeter. Havering uses a single central minicomputer linked by Post Office lines to terminals in the branches (Fig. 7.). The SWALCAP system uses a mini in each library linked by Post Office lines to a central mainframe computer at Bristol. There are useful diagrams in references (10) and (11).

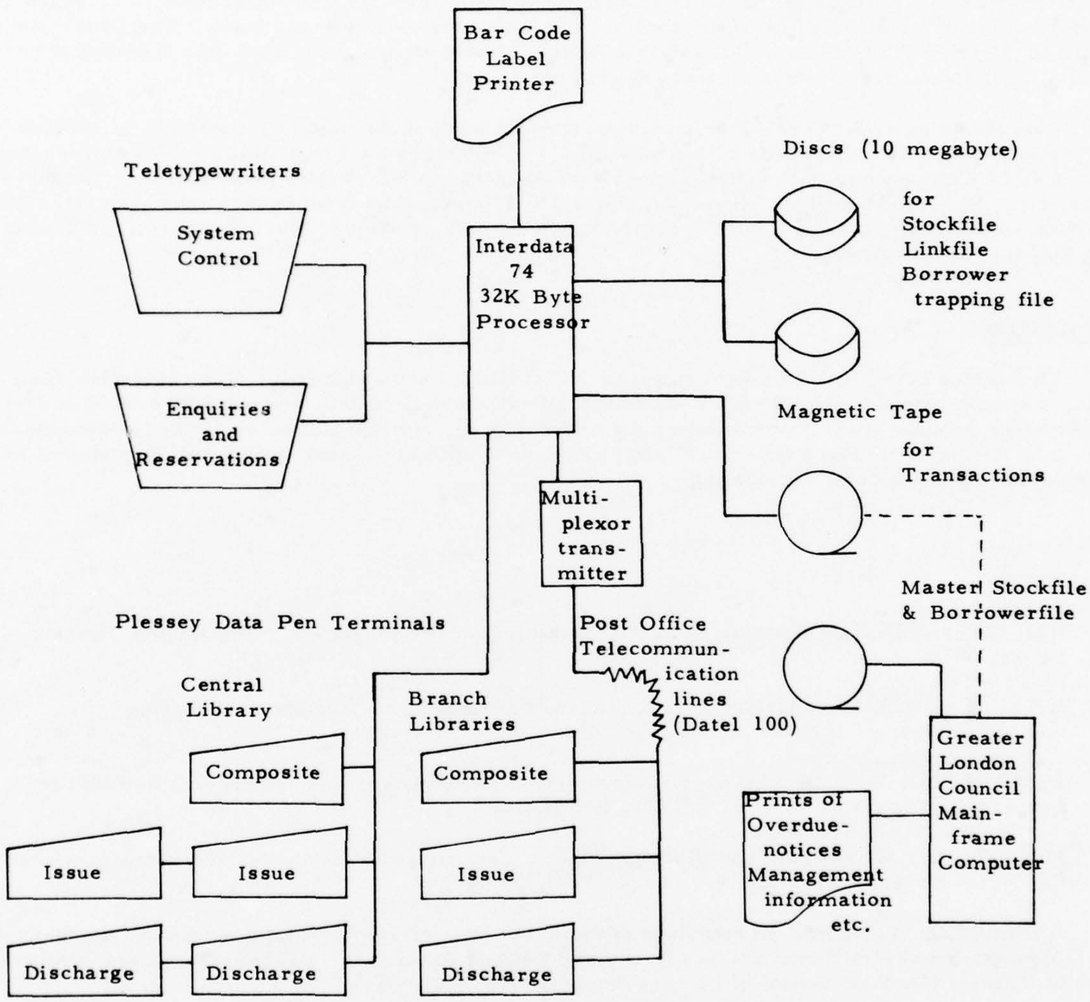


Fig. 7 Havering Public Libraries System Outline

In both cases the minicomputer is used for data validation, but only the Interception List File (Trapping File) is kept on the mini in the SWALCAP system, whereas at Havering the entire Stockfile together with a Linkfile of duplicate copies used in the reservation procedure and the Borrower Trapping File are kept on the minicomputer's discs. In SWALCAP the main files are all on the shared mainframe machine which, being on-line via each library's mini, allows immediate updating of all records. Thus, in that system the mini's main role is as a terminal controller, as a trapping store and also in communication with the mainframe. The Havering system makes use of the Greater London Council mainframe computer for weekly updating of the Master Stockfile. This is accomplished with the use of magnetic tape carried by courier. The GLC machine is also used for providing overdue notices, overborrowing notices and statistical information for management purposes. Similarly at SWALCAP overborrowing is checked for by using an off-line batch-processing program.

The Havering system uses a kind of self-addressing file arrangement for its stockfile, the book identifier being directly related to its disc address. The SWALCAP system uses indexed files. It is possible to do this because they use the concept of the 'Popular File'. This is based on the fact that a small proportion of a library's holdings accounts for most of its book issues. Bristol University keeps an Item File of 45,000 records of which about 12,500 tend to be on loan at any one time. Any book issued which was not on the Item File has its data added from the separate Author/Title File. This concept, with the consequently reduced turnover of records within the Item Files means that the indexes need recreating only every two weeks for the two larger files for Bristol and Exeter University Libraries. But the University College of Cardiff Library Item File is smaller and has a proportionately higher turnover and is therefore indexed weekly. The Author/Title Files is re-indexed every two months.

Data input for the two systems is by entirely different means. Both methods are becoming more common. Havering uses the Plessey Data Pen for reading bar-coded labels for both book and borrower identification together with teletypes for system control, enquiries and hardcopy messages. Similar systems using bar-coded labels are appearing elsewhere, the CLSI system in the U.S.A. and Canada for example. SWALCAP uses the ALS (Automated Library Systems) label readers. These read non-magnetic metallic labels which are used to identify the borrower and book. The labels can be fixed in the back of a book in such a way that except for date stamping the book does not need to be opened. VDU's are used for enquiries and system messages.

Both Havering and SWALCAP rely on numeric only input at the time of transaction. Author/Title detail is added from files held on the system. This requires a more stringent check on the input data. SWALCAP for example, uses Modulus-11 checks plus another of their own devising. Modulus-10 checks would not be good enough. However, the combination of short input data lengths, few and rapid file accesses has contributed to a very speedy response in both systems. The normal rate for a book issue is less than a second.

CONCLUSIONS

The minicomputer can be a very powerful tool for Librarians who wish to keep control of their book circulation systems. It allows for versatility in system design and, depending on such things as the size of the library, it can stand alone or be used in tandem with a mainframe or other minicomputers. The degree of control it allows the library staff will depend upon the system design but the potential is already great and the outlook stimulating.

REFERENCES

1. HARVEY, J. ed. Data Processing in Public and University Libraries. Washington, Spartan Books, 1966

2. WILSON, C.W.J. Comparison of U.K. Computer-based Loans Systems. Program, v.3, nos. 3/4, 1969, p. 127-146

3. LINGENBERG, W. Comparison of Computerised Loans Systems in the Federal Republic of Germany. Program, v.5, no.4, 1971, p. 191-203

4. McCANN, L., McGEE, R. and KIMBER, R.T. Comparison of Computerised Loans Systems in the United States. Program, v.7, no. 1, 1973, p. 24-37

5. LANCASTER, F.W. ed. Proceedings of the 1974 Clinic on Library Applications of Data Processing: Applications of Minicomputers to Library and Related Problems. Urbana-Champaign, University of Illinois, Graduate School of Library Science, 1975

6. Papers presented at the LASIE Workshop on On-line Circulation Control Systems, Melbourne, Aug. 1975. LASIE, v.6, nos. 2 & 3, 1975

7. FOIL, P.S. and CARTER, B.D. Survey of Data Collection Systems for Computer-based Library Circulation Processes. J.Libr. Automn., v.9, no.3, 1976, p. 222-233

8. YOUNG, R.C. United Kingdom Computer-based Loans Systems: A Review. Program, v.9, no.3, 1975, p. 102-114

9. GALLIVAN, B. University of Lancaster Hybrid Issue System. Program, v.9, no.3, 1975, p. 117-132

10. HUDSON, R.F.B. SWALCAP On-line Circulation System: Plans and Progress. Program, v.9, no.3, 1975, p. 133-142

11. PARTRIDGE, D.A. On-line Circulation Control in Havering Public Libraries. Program, v.9, no.3, 1975, p. 115-116

12. ASLIN, C.J. Minis in Circulation Control: Some Parameters and Examples. Program, v.10, no.2, 1976, p.47-54

13. GREEN, M.C. SWALCAP On-line System: a Technical Summary. Program, v.11, no.4, 1977, p. 166-176

14. PICKLES, J.S. Input Methods for United Kingdom Computer-based Circulation Systems. Program, v.11, no.2, 1977, p. 47-63

15. WILSON, C.W.J. ed. Directory of Operational Computer Applications in United Kingdom Libraries and Information Units. 2nd. ed. London, ASLIB, 1977

16. McDOWELL, B.A.J. and PHILLIPS, C.M. Circulation Control System. (Southampton University Library Automation Project, Report No. 1), Southampton, University of Southampton, 1970

17. HAMNER, W.G. The Minicomputer and its Use in Library Operations at the University of Maryland. in LANCASTER, F.W. ed. (Ref.5 above)

ACKNOWLEDGEMENTS

I am grateful to Mr. Richard Hudson of SWALCAP and Mr. David Partridge of Havering for spending time with me in discussing their systems.

THE IDRC'S MINICOMPUTER-BASED BIBLIOGRAPHIC INFORMATION SYSTEM

Faye A. Daneliuk
Project Manager, Minicomputer Project
International Development Research Centre
Box 8500
Ottawa, Canada
K1G 3H9

ABSTRACT

Beginning in January 1978, the International Development Research Centre expects to be operating a minicomputer-based information system to process bibliographic data, to provide many automatic procedures for managing IDRC's library needs, and to permit retrieval from several large data base . The system will be implemented on an in-house minicomputer--a Hewlett-Packard 3000 Series II.

The IDRC's system is sufficiently generalized to permit the creation of data bases of many different types, for example, data about projects, bibliographic data, library accounting data. The concept of data definitions is an integral part of the system software, and it is this which gives the system its flexibility--the system can, in fact, process any data that can be broken down into sets of defined elements.

For over three years the IDRC has operated ISIS (Integrated Set of Information Systems), a software package which was developed by the International Labour Organization to run on IBM 360/370 computers, which processes bibliographic data, provides both batch and on-line (interactive) retrieval capabilities, and has built-in library management routines. The IDRC minicomputer-based system embodies many ISIS concepts but it has been designed in such a way as to conform - as much as it can given the nature of the data - to the relational model of data bases. In addition, a fully interactive user language has been designed which enables the user to do anything from data entry to multi-parameter retrieval or, for example, from generating a KWIC index to producing an accounting of outstanding book orders.

The system has made use of manufacturer-supplied system software, such as the file system, the text editor, and the sort/merge software. To minimize maintenance of the package, no modifications have been made to this software.

The system is small and stand-alone; it is designed for implementation in areas where an inexpensive facility is required.

This paper discusses the basis for acquiring an in-house minicomputer facility and the criteria used for the selection of the hardware. It also presents an overview of the systems design and concepts.

Introduction and Background

The International Development Research Centre (hereafter referred to as the IDRC or the Centre) is a public corporation created by Act of the Canadian Parliament in 1970. It is an autonomous body with a 21-member Board of Governors (drawn from several countries) who set the broad lines of policy and approve individual projects. The Centre's headquarters are in Ottawa, Canada, with regional offices in Singapore, Cairo, Bogota, Dakar and Nairobi.

The objectives of the Centre (in the words of the Act) are "to initiate, encourage, support, and conduct research into the problems of the developing regions of the world, and into the means for applying and adapting scientific, technical, and other knowledge to the economic and social advancement of those regions", and "to help developing regions build up their own research capabilities and the innovative skills needed to solve their problems". In order to carry out these objectives, it was empowered to, again in the words of the Act, "establish, maintain and operate information and data centres and facilities for research and other activities relevant to its objects" and "initiate and carry out research and technical development, including the establishment and operation of any pilot plan or project, to the point where the appropriate results of such research and development can be applied".

Four program divisions were originally established by the Act - health sciences; agriculture, food and nutrition sciences; social sciences and human resources; and information sciences. The information sciences division is rather unique as its establishment marked the first instance in which an aid organization created a program division with the specific objective of supporting information projects in developing countries. It is this division which has involved itself with the application of computers to information work.

In 1972 a project which dealt with computerized information systems was approved by the Board of Governors of the Centre. The purposes of the project were fourfold:

1. to acquire an on-line system which would enable us to computerize our library operations
2. to build a machine-readable data base of our own development literature
3. to work at an international level with other institutions with a view to the development of a cooperative "network" with a "common" system

4. to gain experience which would enable IDRC personnel to aid in the establishment of input/output stations in developing regions.

The system acquired through this project was ISIS (Integrated Set of Information Systems); it had been developed over a period of years by the International Labour Organization in Geneva. ISIS was chosen over a number of other systems, including commercial systems, for some very good reasons: it provided an interactive mode for data entry and retrieval; it provided considerable batch functions for library management; it was international - at the time, the Mexican government office of information and labor was installing it, SAFAD (Swedish Agency for Administrative Development) was using it, both the ILO and UNIDO (United Nations Industrial Development Organization) were using it in Geneva, and the Rumanian government had installed it at the Bucharest Management Centre for Documentalists and Librarians; and it also provided facilities for the exchange of data bases via magnetic tapes which had been formatted according to ISO 2709.

Programmed in IBM 360 Assembler language, ISIS required as a minimum a computer installation running under DOS (Disk Operating System) on a 360 machine. For the IDRC, this meant installing ISIS at a commercial service bureau - acquiring a 360 facility with telecommunications equipment would have meant a prohibitive outlay of funds. The use of a service bureau is not necessarily an inexpensive approach and the project proposal, recognizing this, had left the division with the future option of renting or buying a computer system - either a machine compatible with the ISIS software or one for which we would design new software. In 1975, after 2 1/2 years of ISIS operations, it was decided to investigate the possibility of acquiring an in-house computer.

Acquiring an in-house computer facility

In April of 1975, a consultant was hired by the information sciences division to study the feasibility of implementing a bibliographic information system on a minicomputer. The resulting report, which took into account such factors as costs and software development, was positive in its conclusions and justified more intensive investigation on our part.

In the following eight months a critical evaluation was made of all "stable" minicomputer manufacturers in the market, and of their products. The cost of the equipment, although an important factor, was not the only factor taken into consideration. Because we were intending to do our software development in-house, the extensiveness and reliability of the manufacturer's software was studied in great detail. Our end product - the information system software - was being developed not only for use within the IDRC, but also for use in areas where an inexpensive, reliable facility rather than large capacity would be a requirement. This meant that we had to have some assurance that the manufacturer would remain in business for some time to come. The manufacturer also had to provide some form of service for his equipment in Africa, Southeast Asia, and South and Central America, for we were hoping to make our software available to institutions in those areas. (This latter requirement was not met by any of the minicomputer manufacturers!)

At the same time, we visited other institutions where information systems were being developed for minicomputers. In many of the institutions visited, a machine had been selected to run a dedicated system and our first inclinations were to adopt the same procedure. It can be easily understood that if a machine is dedicated to one application then the manufacturer's software need not be overly sophisticated at all. One can almost accept having to write terminal I/O handlers, file handling systems, and the like if they are to be used in a restricted manner. However, during this period of evaluation, an on-going dialogue was taking place between Information Sciences and other divisions within the Centre. It was seen that the acquisition of a somewhat more sophisticated computer could be of great benefit to the Centre itself. This led to a narrowing of the field of manufacturers who could seriously be considered potential suppliers.

In early 1976, a project proposal dealing with both the acquisition of the minicomputer system and the software development, was presented to the Board of Governors of the Centre by the Information Sciences Division. The project, as approved by the Board, specified three major reasons for going to a minicomputer:

1. To reduce operational costs of running ISIS at IDRC. (This was a significant factor - running costs at a service bureau can be extremely high.)

2. To define an optimum cost-benefit minicomputer installation that could be offered, complete with programs, for AGRIS (Agricultural Information System of the Food and Agricultural Organization of the United Nations)/DEVSIS (Development Sciences Information Systems /ISIS activities at national centres in developing countries.

3. To provide a basis for a Canadian input-output centre to a future international network for development information (DEVSIS).

The three-member "computer group" within the Information Sciences Division drew up the tender which was sent out to a number of manufacturers. The tender stressed three characteristics: a) the power of the operating system, b) the reliability and availability of the manufacturer's software and c) the potential of the machine to handle a job mix. The same group also designed and conducted benchmark tests which were such as to emphasize the three critical specifications in the tender. Hewlett-Packard won the tender with their 3000 system. Although the HP-3000 had some shortcomings which are only now being corrected - specifically, labelled tape processing and support for private disk volumes, some fine hardware features (stack architecture) and their sound, integrated operating system (MPE-Multi-programming Executive) helped them to win the tender. In August of 1976 our equipment was delivered and development began on our own software.

Developing a System

Once a computer was selected, thoughts turned to the system design. At this point in time (March 1976) a final decision had yet to be made that we were indeed going to develop new software for our information system. Other alternatives did exist - we could simply re-code the ISIS programs for the HP-3000 or we could adopt the data base management package developed by Hewlett-Packard for the 3000, IMAGE. The first

alternative, although providing a quick and easy solution to a system design problem, would have proven unrealistic because it could not have taken advantage of the special features of the HP-3000. The second alternative demanded more consideration but was finally rejected for reasons which I am sure are familiar to those of you who have worked with bibliographic systems. Among those reasons were included: 1) no capability for handling true variable length records, 2) no capability for handling variable length, variably occurring fields, 3) no capability for handling subfields, 4) no capability for supporting keys embedded in text and 5) no capability for handling long descriptive abstracts. A new design was definitely called for!

As a first step in designing the system, a number of guiding principles were adopted:

1. General applicability
   - the system should be as general purpose as possible.

2. Modularity
   - the system should be totally modular in order to promote ease of maintenance and extension.

3. Independence
   - the applications functions should be independent of the data base management functions.

4. User considerations
   - the system should be flexible in that it is capable of handling data in almost any physical form.
   - system should be simple to understand in order that it could be implemented and used with minimum effort.
   - a user-attractive language should be provided so that users are really users.
   - the system should be able to provide a wide variety of outputs.

5. Mission orientation
   - the system should have the capacity to accept outputs from other information systems.
   - the system should be viable within a small organization.
   - the system should be compatible with other international systems, specifically ISIS and AGRIS.

6. Cost-effectiveness
   - the basic system should be in operation by December 1977 in order that we could dissociate ourselves from the service bureau where we had been running ISIS.

Theoretical Foundations

In designing a system a set of guiding principles, though very important, are not sufficient. Also required is a theoretical framework around which to build the system. This framework provides a coherence which otherwise would be difficult to realize. Careful study was given to the three prominent data base management theories - the CODASYL (network) approach, the hierarchical approach, and the relational approach. It was finally decided to employ the relational approach in the system's design because this model of data was seen to have a number of inherent advantages not shared by other models.

The relational model is based on the mathematical theory of relations. Although there is a mathematical definition of relations (the interested reader can look at Date, 1975 and Codd, 1970) one can think of a relation as being a collection of unique "flat" records made up of any number of elementary data items. This implies that one can think of a relation as two-dimensional table in which a) no two rows are identical, b) at every row/column position within the table there is only one value, not a set of values (i.e. repeating groups are not allowed), c) columns are homogeneous and d) non-key fields are functionally dependent on the primary key. Given such a structure a generalized set of relational operators are defined for the manipulation of the data at both the domain (or field) level and the relation ("file") level. These operators are defined in the relational algebra (Codd, 1972) and the domain algebra (Merrett, 1976). (Codd also defined a relational calculus but this is left to those readers with greater mathematical knowledge to pursue on their own in Codd, 1971.) An operation executed using the relational algebra is one which takes one or more relations as its operands and produces a relation as a result; the domain algebra does the same thing with domains.

There exist three basic operators within the relational algebra - join, project, and storage. The join is by far the most powerful command available in the algebra; basically it permits you to "put together" two or more relations using a domain as a bond. For example, the natural join (which may be thought of as a generalization of a Boolean AND) of two relations which each have a common domain will result in a relation in which are contained only those records which had identically matching values in the common domain. For example, if you had two relations ACRO and CORP which looked like:

Figure 1

ACRO	ACRONYM	FULL NAME
	ACE	American Council on Education
	ALA	American Library Association
	AACC	American Association of Cereal Chemists

Figure 2

CORP	CORPORATE ACRONYM	CORPORATE CODE
	CIDA	000484
	ACE	004310
	ALA	000062
	ECLA	000573

and, if you executed the natural join of these two relations on the acronym domain your resulting relation would have the following appearance:

Figure 3

ACRO_CORP	ACRONYM	FULL NAME	CORPORATE CODE
	ACE	American Council on Education	004310
	ALA	American Library Association	000062

Five other joins have been defined and can be investigated in the literature (Codd, 1972 and Merrett, 1976).

Projection is an operation which enables one to generate a new relation which is composed of one or more columns of the table which was the original relation. For example, the projection of our newly-created ACRO_CORP relation on full name and corporate code would result in the relation of Figure 4:

Figure 4

FULL NAME	CORPORATE CODE
American Council on Education	004310
American Library Association	000062

Storage (Date, 1975) is a name applied to those operations which effect the insertion and deletion of tuples (or "records"). To insert a tuple into a relation one uses the set union operation - this union produces a set which contains all elements of the original sets. Deletion, on the other hand, uses the set difference operation - the difference will produce a set in which are contained all those elements of the first set which are not contained in the second set.

Domain algebra was, in its practical form, introduced by T. Merrett in 1976. It allows "domains to be the operands and the results of the usual arithmetic, logical and string operators". The operand domains and the resulting domains may be (and usually will be) virtual - they exist only as a defined result which can be actualized on output commands. The operators of the domain algebra may be vertical or horizontal. Horizontal operators work on one or more domains a row at a time whereas the vertical operators work on a column at a time, across all rows. For example, the total cost of all books ordered by a library would be the result of a vertical operation, whereas the number of days taken by a supplier to fill an order would be the result of a horizontal operation - date book received minus date book ordered.

Within the confines of the relational model, the basic "data containers" are the relations. An individual making use of the data base (call this person a user) interfaces with these relations through a data submodel. A data submodel enables the user to see his own view of the data contained within the data base. The submodel is defined using a data definition, and is usually the result of operations (on a relation or set of relations) which were executed by the data base management routines at the request of an application program. Although all three theoretical approaches to data bases management provide some facility for the "redefinition" of data, only the relational model provides a uniform interface for accessing the data at all levels.

A Practical Realization

In any implementation of a data base system, the definition of data is critical - if data is not well defined, it cannot be used. The data must have a structural definition, and its relationship to other data in the system must be understood before it can be used. This information is essential to the end-users of the data base, to the data base management system itself and to the interface between the two. We therefore have three "views" of the data - an "external" view which is seen by the user, an "internal" view which is seen by the operating system, and an intermediate view which relates the external and the internal views to one another. Within the IDRC's system a data definition facility was implemented to deal with this problem.

Four operational levels have been identified for the IDRC data base system:

1. End-user: the end-user of our system may be a researcher, a librarian or a casual visitor.

2. System-manager: this person is akin to the data base administrator who is so often discussed in the literature.

3. Programmer: this is the applications programmer who writes the code which serves as the interface between the end-user and the data base management system.

4. Data base management system: this level sits just above the operating system. All physical access to the data base is controlled by the routines at this level.

Let us start with the end-users. The user will define a data submodel by providing to the system manager the following information: 1) the *name* by which the submodel is to be known 2) a description of the fields of which the submodel is comprised; this description includes the length of the field, the names of the field, an indication as to whether the field is repeatable, field type (numeric or character), and an indication of whether the field could be hierarchically processed, among other things 3) a specification of a default display format 4) an indication of whether any "fast" access paths are required for this submodel, and the fields on which they would be realized.

This information is reprocessed by the system manager to produce an intermediate view of the data for the system. Questions answered at this stage are: 1) does this data exist elsewhere 2) is the date volatile 3) are "fast" access paths justified for the submodel 4) will access be read only, or read-write. (The system manager must take care in defining this intermediate view because the independence of the data within the system must always be maintained.) At this point the data definition processor is invoked and the system manager enters the definition into the system.

At some later time when the user wishes to access the submodel, the data base management system processes the data definition and presents to the application program, the user view, and to itself, the internal view. (Keep in mind that many different relations may be used to represent the data submodel as defined by user. These relations in themselves may be actual or virtual, and may or may not represent a single physical file!) The internal view is always in terms of relations and domains.

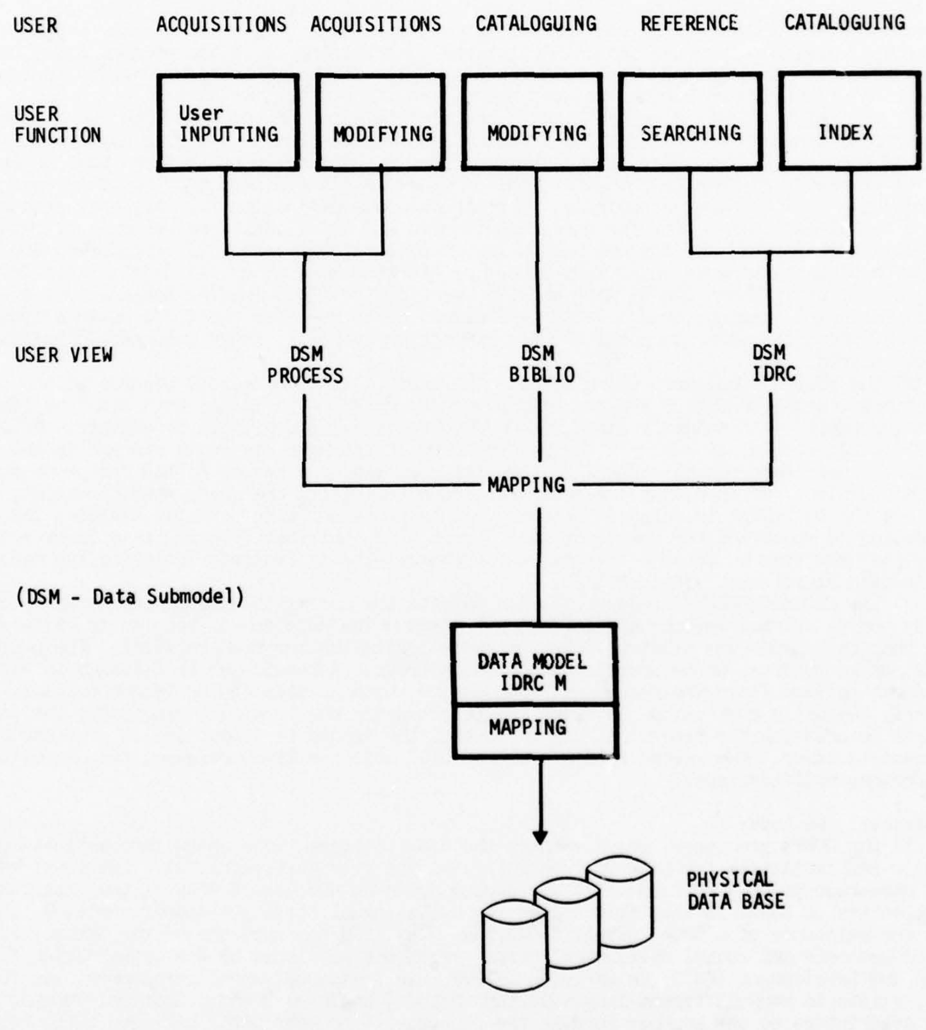


Diagram illustrating the logical path from the end-user to the physical data base.

Having provided a tool for dealing with the various forms of data, we may now look at the functions provided at each level for handling the data. At this point, it is necessary to introduce four concepts which may not be entirely familiar to the reader. They are in order of use:

1. Processor - a processor is an application routine which implements a particular function.

2. Process - a process is the unique execution of a program by a particular user at a particular time.

3. Stream - this term is, to the best of my knowledge, unique to the HP-3000 system. Streaming spools batch jobs (or data) during either interactive sessions or jobs. The spooled job is then scheduled by the operating system, and runs <u>independently</u> of the session or job which streamed it; control returns to the "streamer".

4. Restrict - to restrict means to select a subset of a relation on the basis of certain criteria, the criteria usually being the values held by particular domains. To search for all those records in a data base in which the affiliation is the Ford Foundation is to do a restriction of the data base on affiliation.

The functions at each operational level will be discussed.

<u>End-User</u>

For the end-user, an attempt was made to provide a simple yet powerful language which would make it possible to get the full benefit of the system. The user's requirements were divided into two categories - data entry and data retrieval. Our experience with an automated system had made us aware of problems which were often overlooked; for example - record numbers can be a nuisance, duplicate records can cause a lot of headaches in an environment where gift material can form as large a part of a collection as can purchased material, standardization of data items causes retrieval problems, determining costs can present a tedious manual task, keeping track of bibliographic levels is often useful. It was, therefore, decided to implement six major application processors to handle the user needs. For data entry, we have:

1) the INPUT processor, which permits the user to enter new records into the system. Depending on the structure of the data to be input, an internal sequence number (ISN) may or may not be required for that record. If one is required then it will be generated automatically by the data base management routines. The processor is then told whether bibliographic levels are necessary for this user view; if they are, prompting will be done accordingly. Through the data definition facility this processor can also be instructed to automatically check for duplicate records and to validate the contents of data fields against standard authorities. If the new record is a duplicate, the user will be informed and will be given the option of continuing to enter that record or starting on another; if a data field is not validated the user again will be informed and will be able to re-enter the field (having optionally scanned the authority at this point) having found a valid replacement or to re-enter the field having first been passed to a son process where the user was permitted to generate an authority entry which would validate the field contents just entered.

2) the MODIFY processor, which permits the user to make changes to records within the system. The modifications operate on fields and can be any one of: CHANGE - to change data within a field, ADD - to add a new field, DELETE - to delete a field, TRANSFER - to transfer one field to another, REPLACE - to replace an old field with a new one. If the record on which the user wishes to operate is not accessible by ISN, or if the user does not have the ISN, then the user may do a search within the processor to find the record in question. If more than one record is found to fulfill the query specifications, then the user may select the one which is actually required. This processor also works at a global level; when global processing is specified the processor will stream a copy of itself to run in a lower priority queue, thus leaving the user free to run the terminal for whatever else is desired, including the making of changes to individual records, within MODIFY.

3) the RELEASE/DELETE processor, which permits the user with appropriate security clearance within the system to release records and to delete records. The deletion of records is self-explanatory; at the time that this paper was written, deletions were logical rather than physical. The purpose of releasing is not so obvious, hence an explanation is in order. A record may be released in either of two ways. The first release flags the record as being one to which changes can no longer be made. This feature is particularly useful in a situation in which the person doing the terminal work is not the person filling out worksheets or creating the record; it signifies that the record is clean, and it prevents modifications from being made in error. The second type of release will undo the first release, i.e. it will free the record for further modifications.

For data retrieval, we have:

1) the QUERY processor, which enables the user to search data bases interactively or in "batch". This processor has facilities for both Boolean retrieval and free text retrieval. Retrieval may be done on the full data base or on a portion of it, depending again on the user's view of the data base. An interesting feature of QUERY is that it provides for multilingual thesaurus-aided retrieval. This feature is based on the existence of a true, stable thesaurus. The IDRC has made use of the "Macrothesaurus: A basic list of economic and social development terms" which was published by the Organisation for Economic Co-operation and Development (OECD) in Paris in 1972. This thesaurus, whose development was funded in part by the IDRC, exists in many different languages (among them, English, French, Spanish, German, Portugese), and relates descriptors to one another through the concepts of broader term, narrower term, related term, any term, use term, use for term, and facet number. At the IDRC, descriptive abstracts for documents are written by using descriptors, selected from the macrothesaurus, as embedded keywords. The macrothesaurus exists in machine-readable form at the IDRC in three languages - French, English, and Spanish. QUERY permits one to search in any one of the three languages as a base language, and will translate the

descriptor (automatically OR'ing the translations to the original descriptor entered) into the other languages. For our users at the Centre this has proven particularly useful because many of the documents in our collection are in languages other than English. Furthermore QUERY enables one to search using the structural relations in the thesaurus (for example, BT development aid, will result in a search being done on development aid and its broader terms, in all available languages (if the translation facility has not been disabled by the user)) to any level desired, and to display the structural relations in the base language. Other features include both browsing (sorted or unsorted), and root searching capabilities.

2) the INDEX processor, which enables a user to generate outputs in any wild and fantastic sequences which may be desired. INDEX works on the output of a query or on a user view to produce, essentially, sort keys which will be used to produce ordered outputs from the inputs. This processor is one of the most powerful in the system and, like MODIFY and QUERY, can stream itself to a lower priority queue. It will accept specifications interactively or from a predefined file of specifications, and will also initiate the print job which PRINTs it. It is difficult to discuss this processor on paper and more attention will be given to it during the presentation.

The PRINT processor does all the printing for the system. PRINT will print the outputs of INDEX and QUERY, and will print any user view defined for the data base. PRINT is very flexible; printing can be done on special forms or plain paper; it can be tabular or columnar; it can be with or without diacritical characters, page numbers, and the like; it can be output to a terminal (hard copy or CRT), a line printer, or other device. The printing specifications can be predefined and saved, or defined at the time of the run, through the use of a "chatty" dialogue within PRINT itself. This dialogue obtains from the user all the information required to run a print job, including the field formatting specifications. This same facility allows a user to modify an existing specification. Like the other processors, it is streamable. An arithmetic interface is provided to enable the user to print accounting-like reports; this is printing with computation.

With respect to the end-user, data may be stored in both upper and lower case, and encoding for diacriticals may be embedded with the data (the encoding is recognized by the relevant processors). The working character set of the system is 7-bit ASCII. An arbitrary working maximum record length (data portion) of 2048 characters has been set; it appears to be more than sufficient for current needs. It is to be noted that simultaneous access to the data is a feature of our data base system because of the file locking-unlocking capabilities provided by the file system of MPE. The lock is granular - it is at the relation (file) level as opposed to the record level. (Studies of current literature, and our own experiences, show that this granularity is not detrimental to performance.)

System manager

At this level a number of processors for maintaining and extending data bases have been provided. We have:

1) the ISOCONV processor, which produces and accepts tapes in ISO 2709 format. This is the processor which is used to load data bases received from other organizations. It was made as general as possible, with facilities for special processing exits, in order that we could accept data from as many organizations as possible.

2) the data definition processor, which is used to create, modify, and delete all forms of data definitions (user views, system views and intermediate views).

3) the RENUMBER processor, which permits the replacement of ISN's with other ISN's. This processor is particularly useful in the production of printed bibliographies.

4) the INVERT processor, which does batch inversions on fields in order to provide new rapid access paths to a data base.

5) the "garbage collector", which is periodically run to recover unused and freed space within the data bases, and to generate backups of the physical files of which the data base is comprised.

6) an initialization processor which initializes areas where data bases are to reside.

7) a number initializer to set ISN counters for different user views.

Also available to the system manager are all the facilities provided by the manufacturer. In the case of the HP-3000 we are fortunate to have many.

Programmer

The applications programmer is the person who writes the code to implement the end-user and the system manager processors. Available to this individual are a set of calls to the data base management system. The functions which were made an intrinsic part of the data base management routines make it very easy for the programmer to do things such as permit queries within the MODIFY processor. The functions available to the programmer for the writing of applications processors are procedures which will:

1) perform syntax analyses of user dialogue.

2) actualize virtual domains according to the rules of domain algebra. For the application programmer, this call is made for arithmetic processing.

3) perform restrictions on the data base, and can be used wherever restrictions are desirable. For example, QUERY is a sophisticated interface between the user and the restriction module; INPUT uses it in checking for duplicates; MODIFY uses it to process user queries.

4) project new relations from existing relations.

5) join existing relations to produce new relations.

6) write user records to the data base.

7) read records from a data base.

8) validate the authenticity of data elements (this is used during INPUT, for example).

9) provide the programmer with the capability of reading, writing, deleting, and updating fields within user records.

10) make data bases available for processing, and give to the application the appropriate user view with which to work (i.e., open the data base).

11) close a data base.

12) format records into displayable forms. This procedure is a major component of PRINT; it is also used by QUERY and MODIFY.

13) generate an ISN for a new record entering the system if an ISN is required.

14) extract keys from data elements. Keys may be defined as words, phrases, thesaurus terms, etc., and can be generated in many different forms.

Again, available to the programmer are functions for which provision was made by the manufacturer, for example, SORT/MERGE, and TRANSLATE from EBCDIC and ASCII.

Data base management system

At this level are all those procedures which are our implementation of the theory. The procedures include the fourteen listed above, in addition to three others whose functions are 1) to process generated keys 2) to do the actual work involved in releasing and deleting records 3) to massage data definitions to produce the user view for the application program and the system view for the data base management system (this is called by (10) above).

Although many of the procedures at this level are activated by applications programs, they are also activated by each other to perform many different functions. One reason for this is that at the data base management level, all the routines work with relations and domains, be they virtual or actual. In order to produce a user record to pass back to the caller, GETUPLE (the procedure invoked by the application to read a record) may have to invoke JOIN in order to put together tuples from the constituent relations. Symmetrically, using PROJECT, AUGMENT may have to split the user records into n n-tuples belonging to different relations. At times, GETUPLE may have to use PROJECT to provide a user record.

As explained earlier, it is possible to define user views which are comprised of virtual domains. At the time the user tries to access this view, the procedure which opens the relations recognizes that this is a virtual relation, and activates a procedure which will actualize this relation for the user. All other processing of this relation is identical to that for other relations.

A comment should be made on access. Relational theory does not provide for non-relational access paths to a data base but we have done so. Fast access paths are provided by means of inverted files. Inversion is done on keys extracted from data fields which have been defined for this type of processing by the system manager. The inverted file for a key which is part of an existing data base may be updated at three different times - when the record is first written to the data base (in this case AUGMENT looks after the calling of the key routines to do the key update), when the record is rewritten if the data field containing the key has been modified in any way (in this case the field manipulation procedure looks after it), or when a record is released or deleted (here the data base procedure which actually releases or deletes records does the work of calling the key routines). The three different times are actually two cases - on-line or immediate update and update at release time only. If a record is deleted then all references to it in all inverted files are immediately deleted. A storage technique known as B-trees (Knuth, 1973) was used to implement the inverted files; compacted bit maps were used to store postings and to implement the Boolean logic.

Comments on the Implementation

The system as implemented did not require modifications to the manufacturer-provided software. In fact, some of the facilities which have been implemented were possible only because of the manufacturer's software and hardware. The HP-3000 is a machine which boasts of a stack architecture. This has meant that all our code is reentrant and recursive. The system software is device independent so our software is device independent; processes can be easily spawned through simple calls to the operating system (do you remember our description of the INPUT processor spawning itself as another process); user processes can spawn, as processes, even manufacturer processors; all terminal handling is done by the manufacturer's software (unlike ISIS where terminal I/O handlers had to be written using the PIOCS - physical input-output control system - provided by IBM); processors which we have written can run in any queue, thus they can be streamed, they can stream themselves, and they can stream one another - this is a very important feature because processors which are normally used in interactive mode can be run in "batch" mode if the type of work to be done warrants it; access security is provided by the operating system. It is important to note that the facilities of which we made use are basic facilities - inherent in the operating system software.

All code for the system (application routines and data base management routines) was written in the systems programming languages of the HP-3000 - SPL. This language is a high-level ALGOL-like language which is also the assembler language on the system. Thus it provides one with high level structured statements such as DO....UNTIL and WHILE....DO, yet it also has instructions for bit manipulations. It is doubtful whether our programming efforts would have been as productive if we had had to write in some other language.

Our team consisted of 2 persons for 12 months, and 3 persons for an additional 7 months. During the first 12 months the design of the system was completed, and thirteen of the seventeen data base management routines were completed. During the next seven months, we wrote three of the four remaining data base routines and the user processors. System development began in June of 1976. In October 1977 our first users were phased in (the data entry processors and the print processors were complete) and at the time of writing, we are in the final stages of phasing in our other users. The INDEX processor is complete and the QUERY processor is in the final testing stages. Needless to say, all the system manager processors are operational as well. Our progress is proof that a small, devoted development team is as effective, I would say even more effective, than a larger group.

The data base of the Centre contains some 26,000 references; we also hold the data base of the ILO which runs to over 60,000 references, the data base of the FAO which is in the order of 40,000 references, and two smaller data bases which together are in the order of 10,000 references. Until we can afford to acquire more mass storage, we handle the problem of on-line access through scheduling - the non-IDRC data bases are available at certain fixed times during a week or on advance user request.

Our basic system is operational but our work is by no means complete. Within the next month, the procedure which implements the domain algebra must be written and implemented. By July of 1978, we expect to have implemented an SDI (selective dissemination of information) processor, and a photocomposition interface in PRINT. We also hope to write a procedure which will use a magnetic tape unit as a virtual mass storage unit for the processing of large data bases. We have also to complete the documentation of the system.

It is hoped that this system will prove to be attractive to institutions that require a bibliographic data base system, but cannot afford expensive equipment and have no means of sharing a larger computer facility. It is a logical outgrowth of our experience with ISIS and will, hopefully, help to enlarge the common network which began with ISIS.

References

CODASYL Data Base Task Group Report. ACM, New York, April 1971.

Codd, E.F. (1970) A Relational Model of Data for Large Shared Data Banks. CACM 13, 6, 1970, 377-397.
(1972) Relational Completeness of Data Base Sublanguages. Database Systems, R. Rustin, ed. Prentice Hall, 1972.

Date, C.J. (1975) An Introduction to Database Systems. Addison-Wesley, 1975.

Eswaran, K.P., Gray, J.N., Lorie, R.A., and Traiger, I.L. The Notions of Consistency and Predicate Locks in a Database System. CACM 19, 11, 1976, 624-633.

International Organization for Standardization. ISO-2709 Documentation - Format for bibliographic information interchange on magnetic tape. 1973.

Knuth, D.E. (1972) The Art of Computer Programming, Volume 3/ Sorting and Searching. Addison-Wesley, 1972.

Merrett, T.H. (1976) MRDS - An Algebraic Relational Database System. Proceedings of the Canadian Computer Conference, Montreal, May 1976, 102-124.

Organization for Economic Co-operation and Development. Macrothesaurus: A basic list of economic and social development terms. 1972.

Ries, D.R. and Stonebraker, M. Effects of Locking Granularity in a Database Management System. ACM Transactions on Database Systems 2, 3, 1977, 233-246.

PRODUCTION OF AN ABSTRACTS JOURNAL

By

GEORGE W HART, B.Sc, DIC
Principal Scientific Officer
Defence Research Information Centre
St Mary Cray
Orpington
Kent BR5 3RE England

Summary

One of the most valuable of information tools, both for current awareness and for retrospective searching, is an abstracts journal with cumulative indexes. Although computer-based services such as SDI and on-line searching may be used for similar purposes, the abstracts journal still has an important role to play for organisations without ready access to such services, and particularly for centralised information centres serving many sites. Moreover, the input for an abstracts bulletin can be used to provide the basis for an SDI service and eventually a retrospective retrieval system, on a different computer if necessary. This paper first describes the production by mini computers of the abstracts journals of the Defence Research Information Centre and the Technology Reports Centre, both in the United Kingdom. It then discusses the main points that have to be considered when deciding to undertake such an operation.

1. WHY HAVE AN ABSTRACTS JOURNAL?

1.1 There is a great deal of glamour associated with the use of computers to provide on-line retrieval services and a substantial though lesser amount accrues to the production of a Selective Dissemination of Information Service (SDI). Until recently, such services have required the use of main-frame computers, but mini-computers can now perform these and other tasks in an information centre or library. Applications being described in this Lecture Series include SDI, circulation control, a combined library housekeeping and retrieval system, and the production of an abstracts bulletin, which last is the subject of this paper.

1.2 It might be thought that there is little point in producing abstracts bulletins in these days of readily available on-line services and SDI, and an organisation handling only published items such as books and periodical articles might find it worthwhile to cease announcing new titles and rely on the commercial computer-based services such as INSPEC's SDI service for current awareness and the Lockheed or SDC on-line systems for retrospective retrieval. But I suspect that the cost would be prohibitively high. As an example, an INSPEC personal profile costs about *£100 ($175) a year (one would be needed for each individual or small project group) and on-line searching costs about £20 ($35) per search per data base, see for example refs 1 and 2. For on-line searching there is also a capital cost of at least £1000 for each terminal and modem required.

---

1. Williams, P.W. and Curtis, J.M. The Use of on-line information retrieval services, Program Vol 11 No 1 Jan 1977 pp 1-9

2. Johnston, Susan M, and Gray, D.E. Comparison of manual and on-line retrospective searching for agricultural subjects, Aslib Proceedings, Vol 29 No 7 July 1977 pp 253-258

*Standard profiles are only £40 ($70) a year, but they would probably give a much lower precision for the majority of users.

1.3 Moreover, among the information services commercially available on-line, only two, NASA STAR and NTIS GRA, contain details of report material; and Defence or other limited reports are never available through the commercial services. For an organisation such as a research centre needing to obtain details of report material there seem to be 2 alternative ways of supplying information to the end user:

(a) to develop an internal SDI and/or on-line service, or

(b) to produce an abstracts bulletin.

I believe that a third course is generally preferable, namely to produce an abstracts bulletin by computer and use the data generated to run an SDI service (not necessarily on the same computer) to serve those users whose interests cut across the classification system used to order the items in the bulletin. Of course, if only a small number of items are listed in each issue, there is no need for SDI. However, the same data can also be used to build up a data base for later retrospective searching, whether or not SDI is run.

2. WHY USE A COMPUTER?

2.1 Many bulletins are produced by traditional methods involving typing or type-setting, proof-reading and correcting. But if a bulletin is to be of value for manual retrospective searching as well as for current awareness it is necessary to prepare cumulative indexes to it. These should cover all the different ways in which users are likely to search, either to identify a number of *reports covering a specified subject area, or to locate a specific, known, report. Indexes that might be necessary include subject, author, originator (corporate author), monitoring agency, title, keyword from title, conference and report number. The last item can include the originator's reference number and the information centre's accession number, as well as any other references assigned to the report, eg by the monitoring agency.

2.2 To prepare such indexes by hand is possible but is an extremely labour-intensive and time-consuming task. Essentially it requires the manual preparation of a slip or card for each index entry for each report, each slip or card containing all the details which will appear in the relevant index. The different types of slip must be sorted by hand into the order required, or filed carefully in that order immediately they are produced, and the index copy prepared. Proof-checking of the copy is essential. If some of the indexes are also to appear in each issue of the bulletin, the corresponding slips or cards must be sorted for the issue and merged by hand with the previously cumulated set after each issue has been prepared.

2.3 Using a computer saves all the manual effort since computer programs can extract the required information, sort it and prepare it in the format required for printing. Proof-checking and correction may be needed only once for each report recorded, apart from a relatively quick glance at the computer output to ensure that no unexpected 'bugs' had appeared in the programs thus spoiling the format or the sorting order or missing out or duplicating items.

2.4 Another advantage of using a computer is that if the details of a report have to be suppressed, eg because it has already been announced, or if it has to be placed in a different location in the bulletin, perhaps because the editor decides at a late stage that it is more appropriate to a different subject area, there is no need to leave a blank space, to squeeze the information in, to retype and re-check, or to cut and paste (all possible solutions in a manual system) since the computer can reorganise the material before printing, without manual intervention.

2.5 Perhaps the most important benefit arising from the use of a computer is the ability to build up a computer-readable data base for later use in retrospective retrieval (as mentioned in para 1.3 above).

2.6 There is one main disadvantage. You are tied to a piece of machinery, which will, in the nature of things, break down occasionally. However, provided reasonably quick service is available, delays are unlikely to be much longer than 24 hours since faulty units, including the central processor itself, can easily be replaced. More worrying is the effect of a fire (or deliberate destruction of the computer), and it is essential either to be able to make use of another computer elsewhere or to be prepared to revert to manual methods for a time - if necessary producing an inferior product, perhaps without indexes. Another disadvantage is that special and expensive equipment is needed to prepare the input for the computer, eg a tape typewriter; an ordinary typewriter will not do. Moreover, although computer systems generally allow flexibility, they cannot be changed quickly if new programs are needed.

2.7 The rest of this paper will describe the systems used by two information centres to produce abstracts bulletins by mini-computers. Both evolved from systems which had previously included some mechanisation to produce the indexes for an otherwise manually prepared bulletin.

---

*For simplicity, I shall refer to all items announced in a bulletin as 'reports'.

3. PRODUCTION OF DEFENCE RESEARCH ABSTRACTS AT DRIC

3.1 General Description

3.1.1 Defence Research Abstracts (DRA) is the bulletin of the UK Defence Research Information Centre (DRIC). DRIC is the central facility of the UK Ministry of Defence (MOD) for the acquisition, announcement and distribution of Defence-orientated scientific and technical reports published in the UK and overseas and more particularly those reports which are Defence-controlled or classified. In essence, DRIC is the UK equivalent of USA's Defense Documentation Center and Canada's Defence Scientific Information Service. A short description of the origins and functions of DRIC was given in ref 3.

3.1.2 DRIC receives some 13,000 reports a year of which about 2,000 have been approved for unlimited distribution and are passed to the Department of Industry's Technology Reports Centre after selective distribution by DRIC. The remainder are all indexed and abstracted and some 8,500 are announced in DRA. There are 2 main editions of DRA, a twice-monthly one for MOD and a monthly one for Defence contractors. A second monthly edition with a very limited circulation in MOD is also produced; it is known as the 'B' edition. A quarterly supplement of more highly classified information is also prepared. The balance of the indexed and abstracted reports (about 2,500) are not announced in DRA but appear in a quarterly ('Non-announced') catalogue prepared for the use of DRIC's searchers. These reports are mainly the older ones and those deemed unsuitable by their originators for full announcement. Fig 1 shows the format of an entry in DRA.

3.1.3 All editions of DRA contain subject, author, and report and accession number indexes bound with each issue and printed on distinctive coloured paper. Accumulations containing corporate author (originator), title, conference paper, translation and contract number indexes, as well as the four per-issue indexes, are published quarterly and annually. The annual accumulation for 1976 contained 1450 pages of indexes and was printed as 4 separate volumes.

3.1.4 An extract from the subject index (including a reference to the report shown in Fig 1) appears in Fig 2. It should be noted that provision is made for combining two terms to form a more specific index. 'Access No' is DRIC's serial accession number by which the report is stored in the stockrooms. 'Location' identifies the issue and edition of DRA and the sequential position within that issue (from 2001 for the example shown). The other main indexes are basically similar in form but plans are in hand to print them in double column format, which we believe will be easier to read than the full width lines in use at present.

```
                              1 AERONAUTICS

1-2 AERONAUTICS

[2001-7701]
      BR-54759  RAE-TR-76123                                   UNLIMITED
        Royal Aircraft Est.,Farnborough,Hants.,UK
      PRECISION APPROACH PATH INDICATOR - PAPI
        Smith,A.J.                 Johnson,D.              9.1976     25pp 5ref
                                                               FS-39
      MF AVAIL
      *Landing aids,PAPI(landing aid)/*Guide path landing systems,PAPI(landing aid)/
      *PAPI(landing aid),Aircraft landing/Approach lights/

      The Precision Approach Path Indicator (PAPI) is a simple visual aid that has been
      developed to assist pilots during their approach to landing. It enables pilots to
      acquire the correct glideslope and subsequently to maintain their position on it, thus
      ensuring an accurate approach and landing. Descriptions of two existing systems, VASI
      and T-VASI, are included together with a brief description of PAPI. The operational
      requirements, both current and future, of such systems are discussed, and it is shown
      how the PAPI system test meets these needs. HJB
```

Fig 1. Extract from Defence Research Abstracts

SUBJECT INDEX

Subject Heading & Title of Report	Access No	Location
LANCE(MISSILE)		
FLIGHT TESTS		
Preliminary Data Report (Quick Look) Lance Flight Tests No 278 Missile 3667 Mission NMFT-1	P-215552	2174-7701
Preliminary Data Report (Quick Look) Lance Flight Tests No 279 Missile 3075 Mission ASP 64	P-215640	2175-7701
LANDING		
APPROACH		
Technique for Landing From Steep Gradient Approaches Using a Medium Size Military Transpor	BR-47589	2001-7702
LANDING AIDS		
PAPI(LANDING AID)		
Precision Approach Path Indicator - PAPI	BR-54759	2001-7701
LAP JOINTS		
ADHESIVE BONDING		
Influence of Various Pretreatments on Lapjoint Strength	P-217389	2053-7705
Primary Adhesively Bonded Structure Technology Test Report General Material Property Data	P-216086	2071-7702
MECHANICAL PROPERTIES		
Primary Adhesively Bonded Structure Technology Test Report General Property Data	P-216084	2069-7702

Fig 2. Extract from Cumulative Subject Index to DRA

3. Hart, G.W. The use of a mini-computer at the Defence Research Information Centre. In 'Advancements in Retrieval Technology as related to Information Systems'. AGARD CP-207, October 1976.

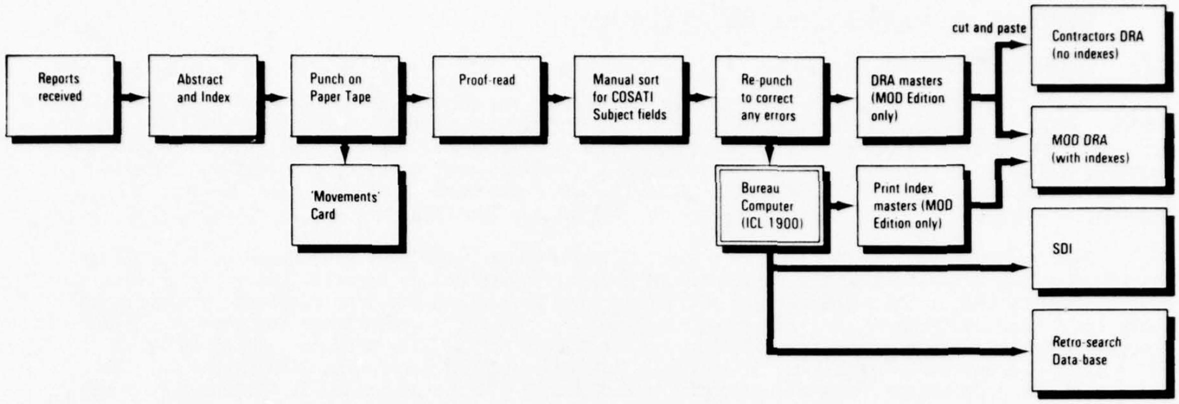


Fig 3a. Production of DRA before 1976 (partially mechanised)

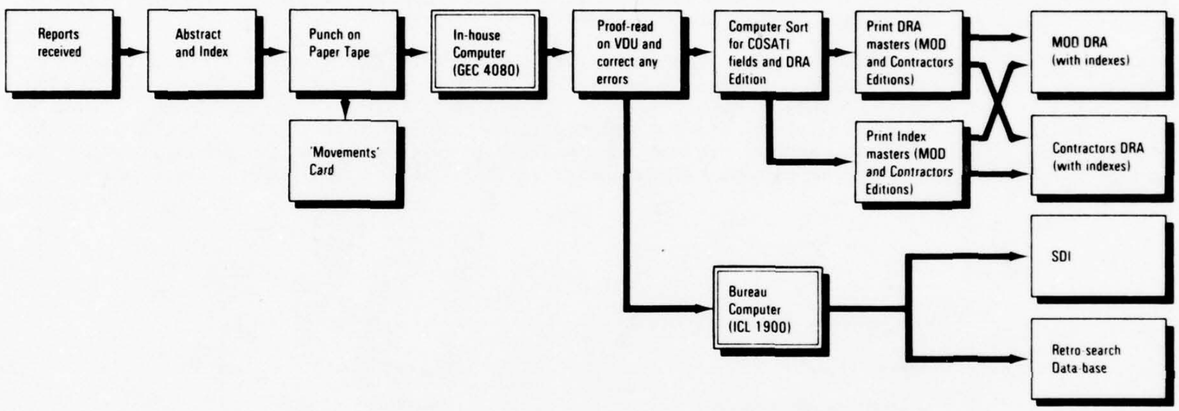


Fig 3b. Production of DRD from 1976 (fully mechanised)

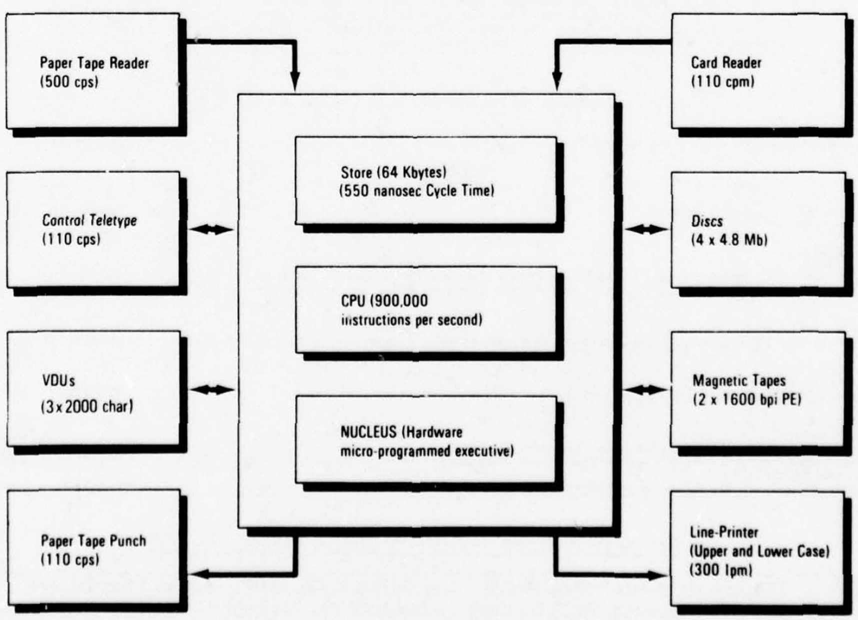


Fig 4. DRIC's GEC 4080 Computer.

3.2 Partial mechanisation

3.2.1 From 1970 to 1975, DRA (and its predecessor, R&D Abstracts) was produced in a partially mechanised manner. This was briefly described in ref 3 and more fully in ref 4. Fig 3a shows the sequence of operations. Essentially, once subject index terms had been assigned and abstracts prepared, the reports were passed to tape typists who punched paper tapes containing these and the bibliographic details, hard copies for proof-checking by clerks being obtained as a by-product.

3.2.2 Errors were marked on the hard copies by the proof checkers, and the copies were then sorted by hand into the order required for DRA (*COSATI Field and Group, ref 5) and returned with the tapes to the typists for corrected tapes to be punched. The typescripts produced during the second typing operation were on A3 paper and were sent to the Reprographic Section of DRIC for the production of DRA by a 33% reduction in size and offset litho printing on A4 paper. The MOD edition was prepared in this way, and the Contractors' edition was prepared from the printing masters by cutting and pasting. The 'B' edition was prepared by re-typing from the corrected tapes.

3.2.3 The corrected paper tapes produced during the second typing were sent to a bureau computer (an ICL 1900) for the preparation of indexes for each issue and later accumulations. The same data were also used to run an SDI service and were retained to form the data base for a retro-search system. With this system, it was impracticable to provide indexes for the Contractors' edition without further re-typing.

3.2.4 It was clear that although the computer was a powerful tool for the production of the indexes it was not being used in any other way. Moreover, a number of the processes required a large amount of clerical effort, but would be amenable to computer-operation if the production of DRA were to be fully mechanised. For these reasons, it was decided to undertake preparation of DRA wholly by computer. A description of the system developed for this follows.

3.3 Full mechanisation

3.3.1 DRA has been produced by computer since early 1976. The processes used are shown in Fig 3b and the computer's configuration is shown in Fig 4.

3.3.2 The computer used is a GEC 4080 computer. It is arguable whether it should be called a mini-computer since it can have up to 256K bytes of central storage (and more recent models may have 1M bytes) and 280M bytes of disc storage, and its processing power is the equal of many main-frame computers of a few years ago. We generally call it a midi-computer and this term is becoming more commonly recognised, but many larger computers are often called 'minis' in the literature. The principles of DRIC's operation would be equally applicable to many smaller mini-computers. The 4080 is a multi-processing computer and so we can carry out other tasks at the same time, including eventually, we hope, on-line retrospective searching with full abstracts among the details displayed. Many smaller mini-computers would be unable to do this, but could be used to prepare a bulletin in a similar manner.

3.3.3 The first 3 stages shown in Fig 3b are essentially the same as in the partially mechanised system (Fig 3a) with the production of the 'Movements' cards for use by the report handling sections as a by-product. The paper tapes are fed direct to the 4080 computer and their contents stored on disc. The report details can then be displayed on a VDU screen for checking and correction by the proof-checkers. The advantages of the VDU correction are that the person who spots an error also makes the correction, thus making a correct finished result more likely, and no re-typing or further tape-reading are necessary. The last point is important because paper tape readers are notoriously liable to reading errors.

3.3.4 After errors have been corrected, the computer sorts the items into COSATI order and extracts sets of items appropriate to the various editions, assigning location reference numbers at the same time. These numbers are serial numbers identifying the position of the item in each edition. The MOD numbers are 1001 upwards and those for the Contractors' edition are 2001 upwards. The number also indicates the year and issue. Thus 2001-7701 was the 1st item in the Contractors' edition of issue 1 of 1977. This item also appeared in the MOD edition, where it had the number, 1008-7701. The location reference numbers are given as references in each of the printed indexes and so enable the bibliographic details and abstract to be located quickly by someone searching the indexes. The DRIC accession numbers are also given as references in the indexes. They are needed to request reports from DRIC. A magnetic tape containing full details of all the items is prepared at this stage and passed to an ICL 1900 series computer at an MOD bureau for use in the SDI and retrospective search systems.

---

4. Ridler, E.H. et al. Production of R&D Abstracts and printed indexes to R&D Abstracts using tape typewriters. TIL report 21, Aug. 1969.

5. Defense Documentation Center. COSATI Subject Category List (DOD Extended), AD 624000, December 1965.

*Committee on Scientific and Technical Information of the Federal Council on Science and Technology.

3.3.5 The details required for the various indexes for each edition are then extracted to form separate files, four (subject, author and report and accession number) for each issue of each edition, and cumulative files for all the indexes. Finally the material is formatted as required for the various editions and indexes and printed on an upper and lower case line printer at about A3 size for later reduction by 33% and offset litho printing on A4 paper.

3.3.6 The Contractors' edition now has indexes and the various time-consuming manual operations previously required have been replaced by ADP processes. The advantages seen by DRIC are summarised in Fig 5 (below), the main ones being the first five. A further advantage, which will arise only when the complete system is implemented, is the 100% validation of descriptors that will be possible by holding a list of approved descriptors on the computer.

Automatic checking of data

No manual sorting (for subject category or edition)

No re-typing for different editions

No cutting & pasting for different editions

Indexes for Contractors' Edition

Greater Consistency of Corporate Author references

Upper & lower case indexes

Multiple entries of the same item

Less movement of classified material

Own control of processing

Faster turn-round

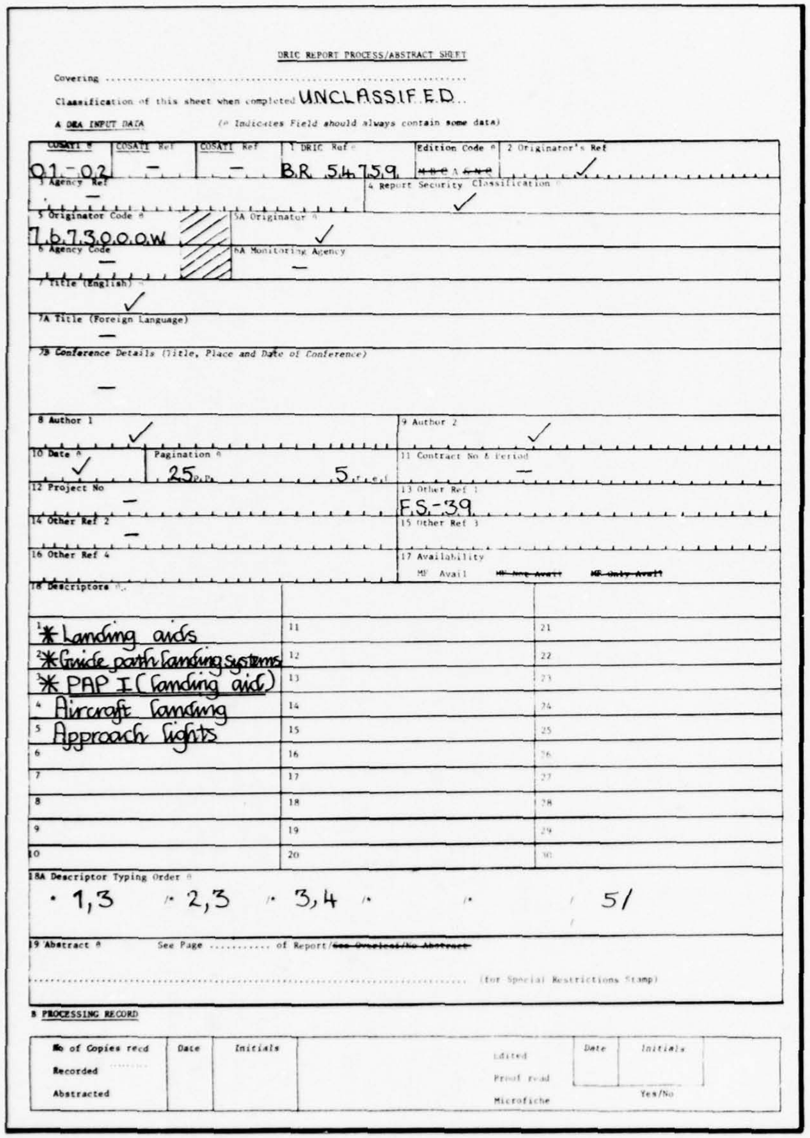
Fig. 5 Advantages obtained from fully computerised production of DRA

3.3.7 Having summarised the production process, I will now discuss the various stages in more detail, describing not only the processes but also some of the problems encountered as I feel sure that the latter are of most interest to other organisations planning to computerise an abstracts bulletin.

3.4 Input

3.4.1 The basic input document is a report process sheet (Fig 6). This contains a number of boxes, each corresponding to a field in the computer record; and different fields are completed by clerical staff, by subject specialists and by the editorial staff. To save time, where the details are found on the cover or title page of the report they are suitably annotated, ringed and marked with the appropriate field number, and the corresponding box on the process sheet is ticked. Where a field is not applicable, the box is left empty. Fig 7 shows the cover of the report corresponding to the process sheet shown in Fig 6 and the bulletin entry in Fig 1.

3.4.2 Use of this form should be clear from a comparison of these three figures. The first COSATI field is the main one and the complete entry will appear there in the bulletin. If additional COSATI references are used, the bibliographic details will be printed at those locations without the abstract but with a cross-reference to the main entry. The Edition Code field contains the codes shown in Fig 8 (next page) and identifies the edition(s) of DRA in which the item will appear. Box 5 contains the code number for the originator, including a check character, and the computer uses this to look up a disc-based list of originators, thus ensuring that the references to a given organisation always appear in the same form. Box 18A is used to indicate the arrangement of the descriptors in the subject index, those marked with an asterisk appearing as entries and sub-headings in that index (see Fig 2). The descriptors not marked with an asterisk do not appear as entries in the index but are available for use in SDI and retrospective searching. The numbers in box 18A of course refer to the numbers of the descriptors in Box 18.



DRIC REPORT PROCESS/ABSTRACT SHEET

Covering ....

Classification of this sheet when completed UNCLASSIFIED

A DRA INPUT DATA (* Indicates Field should always contain some data)

COSATI *	COSATI Ref	COSATI Ref	1 DRIC Ref *	Edition Code *	2 Originator's Ref
01-02	-	-	BR 54759		✓

3 Agency Ref: -

4 Report Security Classification: ✓

5 Originator Code *: 7673000W

5A Originator *: ✓

6 Agency Code: -

6A Monitoring Agency: -

7 Title (English) *: ✓

7A Title (Foreign Language): -

7B Conference Details (Title, Place and Date of Conference): -

8 Author 1: ✓

9 Author 2: ✓

10 Date *: ✓

Pagination *: 25 pp 5 ref

11 Contract No & Period: -

12 Project No: -

13 Other Ref 1: FS-39

14 Other Ref 2: -

15 Other Ref 3

16 Other Ref 4

17 Availability: MF Avail ~~MF Not Avail~~ ~~MF Only Avail~~

18 Descriptors *

1 *Landing aids	11	21
2 *Guide path landing systems	12	22
*PAP I (landing aid)	13	23
4 Aircraft landing	14	24
5 Approach lights	15	25
6	16	26
7	17	27
8	18	28
9	19	29
10	20	30

18A Descriptor Typing Order *

1,3 2,3 3,4 5/

19 Abstract * See Page .... of Report/~~See Overleaf/No Abstract~~

.... (for Special Restrictions Stamp)

B PROCESSING RECORD

No of Copies recd	Date	Initials		Date	Initials
Recorded			Edited		
Abstracted			Proof read		
			Microfiche		Yes/No

Fig 6. A Report Process Sheet.

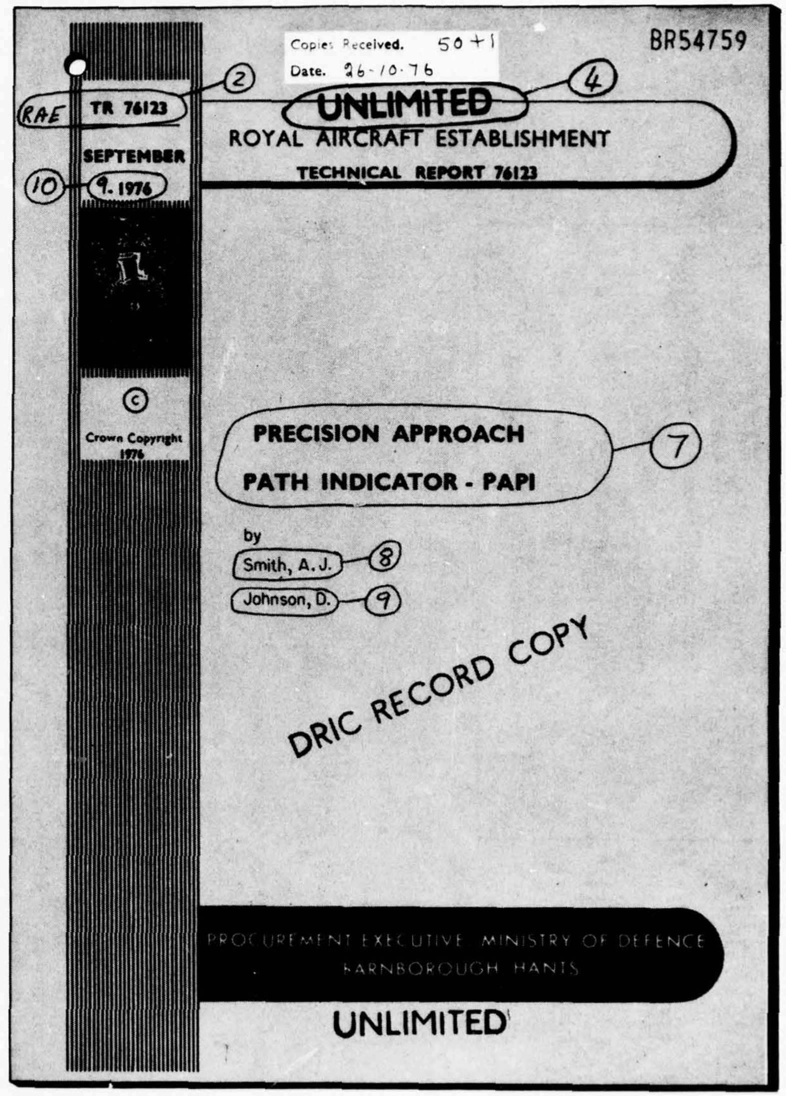


Fig 7. A Report Cover Annotated for Data Input.

M	MOD Edition (twice monthly)
B	B Edition (monthly)
C	Contractors' (monthly) and MOD Editions
A	All above
S	Supplement (quarterly)
N	Non-announced catalogue (quarterly)
Q	Query (edition to be decided)

Fig 8 Codes for editions of DRA

3.4.3 The report and its associated process sheet are used by tape typists who punch a paper tape with all the details. Each field is terminated by a separator (∇) and blank fields are indicated solely by their separators. The format produced is similar to that which will appear in DRA and is controlled by a control tape read on a second reader.

3.4.4 The details of conferences (7B) need to be input in a standard form if the conference paper index is to be satisfactory, ie not to have two or more variant forms for the same conference. This is done by preparing a short length of paper tape containing the conference title, location and date and inserting this in the tape typewriter reader when required for each individual paper abstracted.

3.4.5 The details of about 30 reports, including their abstracts, are punched on each reel of tape, this being a convenient size for handling. During the partial-mechanization stage, however, the first typing produced single lengths of tape for each report and they were put into envelopes with the type-scripts and sorted by hand. When the GEC 4080 system was first started many hundreds of these tapes remained to be used, and it was found to be an extremely slow task to handle them individually. Consideration was given to putting them through the tape typewriters to produce full length reels, but this would have involved just as much time. However, this problem was responsible, by itself, for quite a considerable delay in the production of DRA. There seemed to be no obvious way round it but it is a point to be borne in mind by other organisations in a similar situation.

3.4.6 It is desirable for issues of DRA to be roughly similar in size and at first we used the same number of reels of tape for each issue. However, the percentage of 'non-announced' material (ie items not being published in DRA) varied from 20% to 50% and this caused the size of issues to fluctuate widely. We could have got over this by handling the non-announced items separately but in fact we wrote a simple program to count the DRA items as the tapes were input, enabling us to use an appropriate number of tapes.

3.5 Checking

3.5.1 Each item is validated by the computer to check for obvious unacceptability. The first check made is that the total number of field separators is correct; if this is not so, the item is rejected. Once accepted, other checks are made, for example that the accession and location numbers are in the correct form, that the date of the report is in the correct form and is a feasible one, that the fields with a maximum length, such as report number and author, are not too long, that an acceptable classification is present and so on. A further check to be introduced soon is that the subject descriptors assigned are on a list of approved terms.

3.5.2 After validation, the items are displayed on a VDU for checking by the clerical proof-checkers who compare them with the corresponding reports and process sheets. An example of the display is given in Fig 9 (next page) the item separators now being shown by '?' and not '∇'. These are in fact different interpretations of the same code by 2 different machines. Although a complete item is shown for convenience, in practice it would be divided into 3 parts, each being shown in sequence. Many items could not have been accommodated fully on the screen, so rather than split them at an arbitrary point, it was decided to split them all after the bibliographic details and after the descriptors.

3.5.3 The marginal annotations in Fig 9 are error messages generated by the computer during validation. The first shows an unacceptable date (a month numbered 15) and a descriptor longer than 40 characters (a '/' having been omitted). The other errors ringed are ones that can only be spotted by a human proof-checker. If the corporate author code is not acceptable to the computer, an error message is shown, but if the number is acceptable although incorrectly assigned there is no error message and so the proof-checker must always ensure that the correct corporate author is present.

```
        01-01
        BR-19224 ?A?ARC-CP-1091? ?
        UNLIMITED?
        027500 Aerodynamics Dept., Royal Aircraft Est., Bedford, U.K.?
        Atmospheric Gusts-A Review of the Results of some Recent
        R.A.E. Research?
        ?
        ?
DATE    Burnham, J.? ?15.1976? 59pp 57ref?   ?
        ?  ?  ?
        ?  ?  ?

        *Wind (meteorology), Gusts, Aircraft/Aircraft, Gusts, Thunderstorms
>40c    /Stratosphere/Flight characteristics Mathematical models/?

        Recent RAE research on guests has been particularly concerned
        with severe gusts and the situations in which they occur. In
        the sratosphere mountain wave conditions and those in the
        vicinity of thunderstorm tops have been investigated. At lower
        altitudes, gusts in and near thunderstorms have also been
        studied, as have wind and gust effects likely to be significant
        during take-off and landing. This work has relevance both to
        aircraft operations and to aircraft design. In the latter
        connection, recent work on mathematical models of severe gusts
        is also described. Mention is made is made of the effects of
        pilot control activity during flight through gusts.  PBP?
```

Fig 9. Example of the Display on VDU for Proof-checking.

3.5.4 Corrections are made by the proof-checker by using the editing controls on the VDU, ie by moving the cursor to the appropriate positions and over-typing, deleting or inserting characters. The VDU automatically inserts new lines on the screen as necessary when characters or even whole fields are inserted, but they are run together before being re-input to the computer so that the output is unaffected. When the checker is satisfied with the display on the screen the item can be re-input to the computer. If the record is too bad to be corrected, the item may be sent to a reject file for later correction or re-typing.

3.5.5 I have described this system of checking and correcting in the present tense although it has not yet been brought into use, since we expect to have it working in 1978. Until it is, the proof-checkers will continue to mark up hard copy, as in the earlier semi-automatic system, and the tape typists will produce a second tape with the necessary corrections.

3.5.6 The number of tapes to be input for each issue of DRA is assessed at present as explained in paragraph 3.4.6, but when the on-line correcting system is in use, we will no longer need to use an integral number of tapes because the computer will store all corrected records in a cumulative file from which it can be told to select the number needed for an issue.

3.6 Processing

3.6.1 Once the items for a complete issue have been selected, the computer sorts them into COSATI field and group order and prints a complete list of all the items in DRA format (the so-called *BLIS listing) which is passed to the Editors for final checking. The main purpose of this listing is to enable the COSATI headings and the editions of DRA to be re-checked. For example, great care must be taken that too highly classified information is not printed in the Contractors' Edition. This is to be checked by computer soon, but certain other types of errors are only detectable by human inspection. At present, any further corrections are made at the VDU by the computer staff using the editing software supplied with the operating system. When the on-line correction system is in operation, the Editors will be able to correct the entries at a VDU in the same manner as the proof-checkers.

*BLIS = Bulletin List

3.6.2 When this final check has been completed, the computer extracts sets of records corresponding to the different editions of DRA. Cross-reference entries without an abstract are generated for any subsidiary COSATI headings. This was possible in the previous system but involved partial repetition of typing and was kept to a minimum. Now it merely requires additional entries on the process sheet. The previously required sorting of up to 400 envelopes containing tape, etc, into subject order, used to take a considerable time when performed manually. The records are now sorted into COSATI order (and accession number within that) very quickly by computer, which also assigns the location reference numbers.

3.6.3 Sorting complete bulletin entries would be a very lengthy process and it is most efficient on the 4080 to sort records no longer than 248 characters. So each entry is divided into 9 record types each containing one or more fields. Any types longer than 248 characters need one or more trailing records. Each record includes a 16 character header comprising COSATI number, Accession number, record type (0-8) and occurrence indicator (A for the first record and B, C, ... for trailing records). This header is used as the sort key.

3.6.4 An alternative method might have been to have retained each report entry as one record and store it on a disc, sorting only the COSATI number and accession number with a reference to the disc location. However, this would have been more complicated to program at a time when all DRIC's staff were unfamiliar with the GEC 4080 and so the simpler approach was adopted. Sorting a complete issue of one edition takes only 5 to 10 minutes so its use of processor time is not too excessive.

3.6.5 Once the sets of items for the various editions have been compiled, the computer extracts the parts of the records needed to prepare the different indexes. Each index set is sorted according to the main and subsidiary entries, eg the subject index is sorted by subject index terms and title. This is easily accomplished by forming each extract into a record with the entries in the appropriate order and then sorting the complete record.

3.6.6 The GEC 4080 computer was originally sold mostly for process control purposes or for use as an interface between laboratory equipment and a main frame computer. DRIC was the first organisation to use one for automatic data processing and so the software available did not include a sorting procedure.

3.6.7 The sort software was written by GEC under contract and seemed to be satisfactory at first. It was only when we looked closely, particularly at the title indexes, that a problem became apparent. The GEC is a byte-oriented machine with different representations for upper and lower case letters. The sort puts the letters into the order A, a, B, b, C, c, etc, but this means, for example, that 'ACE' sorts before 'Acceleration' and not afterwards as might be expected. This is somewhat confusing to the user, particularly when the upper case abbreviation can be read as a word, as in this case. It seems to be less important for unpronounceable expressions such as 'MRCA', perhaps because we are used to telephone directories which put such items at the beginning of each letter. The title index is the only one in which this problem arises since the others such as subject and corporate author have the index entry points in upper case and the sorting is done with the items in that form.

3.6.8 However, the computer is also used to print a complete list of corporate authors and their codes, which is referred to regularly by clerical staff. For that list, the problem of the sort would be much more serious and so the items are all converted into upper case before the sort starts and converted back to their original form afterwards by looking up a list of the entries which is held in the computer.

3.7 Output

3.7.1 Having organised the material in a suitable form, it has to be output. DRIC use a relatively slow line printer with an upper and lower case print barrel and the print-out is used as copy for plate making for printing. The quality of the print-out is quite good but the subsequent 33% reduction in size and offset litho printing appear to accentuate the thicker strokes, making the final product slightly less easy to read than the typescript used previously. A better quality would be obtained if we printed at full size, but this would require nearly twice as much paper, and the bulletin would be less easy to use with fewer items on a page.

3.7.2 We are now considering other ways of printing, eg using a daisy-wheel printer, computerised phototypesetting or a Xerox or similar machine driven by magnetic tape. Ink-jet printing would be another possibility but is likely to be too expensive for us to have the equipment in-house.

3.7.3 We were originally urged to use nylon ribbons for the printer as they are about half the cost of silk ones. But we discovered that silk ribbons give a better quality reproduction and we have continued to use them. The best results seem to come from the use of medium weight paper.

3.7.4 A minor problem seems to have occurred when the printer is unable to print at a steady rate because the lines of print are not ready in time, eg when a number of disc accesses are required. It appears that although the printing paper stops, the ribbon continues to move and occasionally rubs against a fold in the paper leaving a black mark. We have to print across alternate folds because the length of a sheet of printing paper is less than is needed to give an A4 size page after reduction. This has been overcome by rearranging the formatting and print program so that fewer disc accesses are needed during printing.

3.8. Problems in implementation

3.8.1 As with most computer systems, a number of problems arose during the early days of using the mini-computer, although none of them was unduly serious. They are described in some detail in paragraphs 26-32 of ref 3 and the details need not be repeated here. The main lesson learnt was that parallel running of the old and new systems requires almost as much preliminary thought as designing the new system and that system design should not be completed without detailed consideration of the arrangements and the

additional resources required for parallel running. This is probably particularly true if you are converting from one mechanised system to another, but it is also applicable when using a computer for the first time.

3.9 Programming

3.9.1 The GEC 4080 computer supports a number of languages, including CORAL 66 and FORTRAN IV. CORAL, a real-time processing language developed in the UK Ministry of Defence, would probably have been suitable, but we were advised to use GEC's assembler language, BABBAGE, which has many high level features and so enables programs to be written reasonably quickly whilst still maintaining control over characters and individual data bits when necessary. The effort required by DRIC to implement the system was about 4 man years, using two trainee programmers and a team leader with no experience of BABBAGE or the GEC 4080. They wrote all the programs apart from the sort, which was provided under contract by the manufacturer, and the VDU edit, which was a by-product of other work being done by the manufacturer. The latter, incidentally, required a good deal of additional effort by DRIC's staff, but this is not included in the figure above.

3.10 Justification

3.10.1 Why was the system installed and have we obtained the advantages foreseen? A detailed feasibility study carried out in 1973 concluded that the installation of a mini-computer would:

(a) increase efficiency in the preparation of data for DRA and indexes;

(b) increase security in the production of DRA (by reducing the amount of movement of classified information);

(c) increase security in the movement of classified documents (see paragraph 3.10.4 below); and

(d) provide a generally improved service to all DRIC customers.

3.10.2 We have realised benefits (a) and (d) since less typing and clerical effort is required to prepare DRA yet all editions now have indexes (per issue and cumulative). To have provided indexes for the Contractors' Edition under the previous system would have required 2 extra typists, it was estimated. These savings probably do not cover the cost of the installation, let alone the programming team, but having the computer does enable us to carry out other tasks which are referred to below.

3.10.3 It is of course impossible to assess benefit (b) in terms of money but it does mean that classified print-outs of indexes are no longer sent many miles by van. They still have to be taken a short distance to a local printer, as was done with the previous system, because we have insufficient capacity to print them in-house.

3.10.4 We intended originally to obtain benefit (c) by keeping records of the movements of classified reports on the computer but we have recently decided to aim for a more integrated approach in which the details of all reports will be entered into the system when they arrive at DRIC. This will enable all later movements, distribution, and changes of classification to be registered with the minimum of effort, and the same details will be available as the basis of the DRA record. We also use the computer to record the numbers and classification of reports exchanged with foreign countries and this information could be compiled to a large extent from the basic details entered initially.

3.11 The future

3.11.1 The probable addition of a thesaurus check was referred to in paragraph 3.5.1 above. Other plans for the future are the replacement of the tape typewriters by some other form of data entry such as key to disc, the revision of the report process sheet to include additional fields such as the classification of the title and abstract (at present these are given in brackets at the end of the appropriate field, but that is not wholly satisfactory), and an extension of the validation process, eg to ensure that items are announced only in the correct edition or editions. Apart from these changes, we expect to consolidate the DRA system, whilst concentrating on the integrated approach mentioned above and the addition of SDI and retrospective searching to the in-house system. At present, the last two are carried out on a bureau ICL 1900 computer, but that will be replaced by a new machine in 1981 and the system will then need to be re-programmed. Bringing SDI and retrospective searching in-house would increase our control over them.

4. PRODUCTION OF R&D ABSTRACTS AT THE TECHNOLOGY REPORTS CENTRE (TRC)

4.1 Introduction

4.1.1 TRC is a part of the UK Department of Industry and was formed at the same as DRIC (1971) when the previously existing Mintech Reports Centre was split up. It handles unlimited reports produced by Government or Government assisted bodies such as UK Research Associations. Essentially it is the UK equivalent of NTIS. TRC produces an abstracts bulletin, R&D Abstracts, which is similar in format to DRA. There are 2 editions, one appearing twice-monthly with 6-monthly cumulative indexes, the other appearing every 2 months with annual indexes.

4.1.2 TRC also use a British mini computer - a Business Computers (Systems) Ltd Molecular 18. This has 64K bytes of store with one magnetic tape deck and two disc units giving about 40M bytes. They have an upper and lower case printer, an input/output golfball typewriter, and 6 VDUs, each with a 1280 character display, and various other peripherals.

4.1.3 As would be expected from the common parentage, TRC and DRIC's announcement bulletins are very similar in style and were originally produced by almost identical partially mechanised methods. With the advent of the mini computer, however, the paths have diverged and I intend to describe briefly how TRC operate. A full description is given in ref 6.

4.2 Production of bulletin & indexes

4.2.1 The process is shown in Fig 10 (taken from ref 6). The letters that follow refer to that figure. Process sheets, similar to those used by DRIC are completed by the Recording and Information Analysis Groups (A). The details of up to 36 records are then input to two daily input disc files (one for COSATI fields 1-11, the other for fields 12-23), using a VDU as a direct entry terminal (B). Each record is set up by the operator in the format required for announcement in R&D Abstracts. The data entered can be corrected by the operator using the cursor controls and editing facilities of the VDU before being transmitted to the disc file. Each record must consist of 20 fields separated by terminators (?), and of course blank fields must be represented by a terminator. Each operator's work is checked at a VDU against the original documents by another operator.

4.2.2 Each day, sorted indexes to the random disc files are produced (C) and printed in accession number order within COSATI code. The full input records are validated daily (D), some 10 checks being made on the data, eg the number of fields, the format of the COSATI code, etc. An error list is printed and corrections can be made daily by retrieving the records using their accession numbers and editing on the VDU (E). After validation, the daily files are added to the semi-monthly cumulative random files from which the next increment of R&D Abstracts will be produced (F). Again, 2 files are used, corresponding to COSATI fields 1-11 and 12-23. Each may hold up to 180 records. The entries made during the day are printed for checking by the data preparation Supervisor, who can also call up and correct on the VDU any report she wishes.

4.2.3 At the end of a cumulative period, the index records for the whole period are extracted and the bulletin data again pass through the validation program (G). A print-out is produced on the line-printer for the Editor of the Bulletin who checks, inter alia, index terms, COSATI codes and spellings of proper names. Any corrections necessitated by these checks can of course be made at the VDU (H). The body of R&D Abstracts is printed on the golf-ball typewriter in accession number order within COSATI code and entries are selected from the random files by a similarly sorted index, which identifies the location on disc of each record.

4.2.4 Each issue of R&D Abstracts has 5 indexes (subject, author, report number, accession number and COSATI code). The relevant data fields are extracted from the semi-monthly incremental file and set up in random order to form a general index file. To print an index, eg of authors, the first process is to extract the basic information required for ordering the items (authors' names) from this general index file together with a reference to the location in the file of each item, thus forming an internal index to the general index file. The internal index is then sorted (by author) and the full author index is produced from the details in the general index file using the same order as the sorted internal index. A similar procedure is then followed for the other indexes. The advantage of this procedure is that only very small records need to be sorted, thus considerably speeding up processing. The indexes are printed on the line-printer to save time. The extracted index information is added to a cumulative file for later production of the 6-monthly indexes (I).

4.2.5 The body of the bulletin is printed at full size but the indexes are photo-reduced by 30% before printing. Both are printed in-house by off-set litho reproduction.

4.2.6 As at DRIC, a magnetic tape is prepared for use on an ICL 1900 series computer (J). The SDI and retrospective search program suites have been run on a number of large ICL machines but the work is currently concentrated on the Department of Industry bureau computer at Eastcote, Middlesex, where the accumulated data base is held.

4.2.7 TRC also prepares indexes to 'US Government Reports Announcements' using twice-monthly magnetic tapes supplied by NTIS. This requires a code conversion and data formatting process followed by field selection, sorting and printing routines similar to those used for the indexes to R&D Abstracts.

4.3 Photo typesetting

4.3.1 Plans have been made for the production of the bulletin by computer controlled photo-typesetting. This involves ICL 1900 programs that format and translate the output magnetic tape (J) so that it can be used as direct input to a Linotron 505 photo-typesetter. Output is in the form of bromide positive prints for subsequent off-set litho reproduction. Up to 1000 characters can be provided, but TRC need only about 250, comprising roman, italic, bold and capitals. The video optical system produces a high quality image in 15 sizes from 4 to 28 point and lines up to 64 picas wide are exposed onto 100 foot rolls of film or paper. The TRC software allows point size (character height) changes within a line, and set size (character width) to be varied to provide a condensed or expanded type style. TRC have run some sample bulletins using this system but at present its introduction has been delayed by factors beyond their control.

---

6. Adams, H.C. Computer processing of bibliographic data at the Technology Reports Centre of the Department of Industry, TRC, 1977

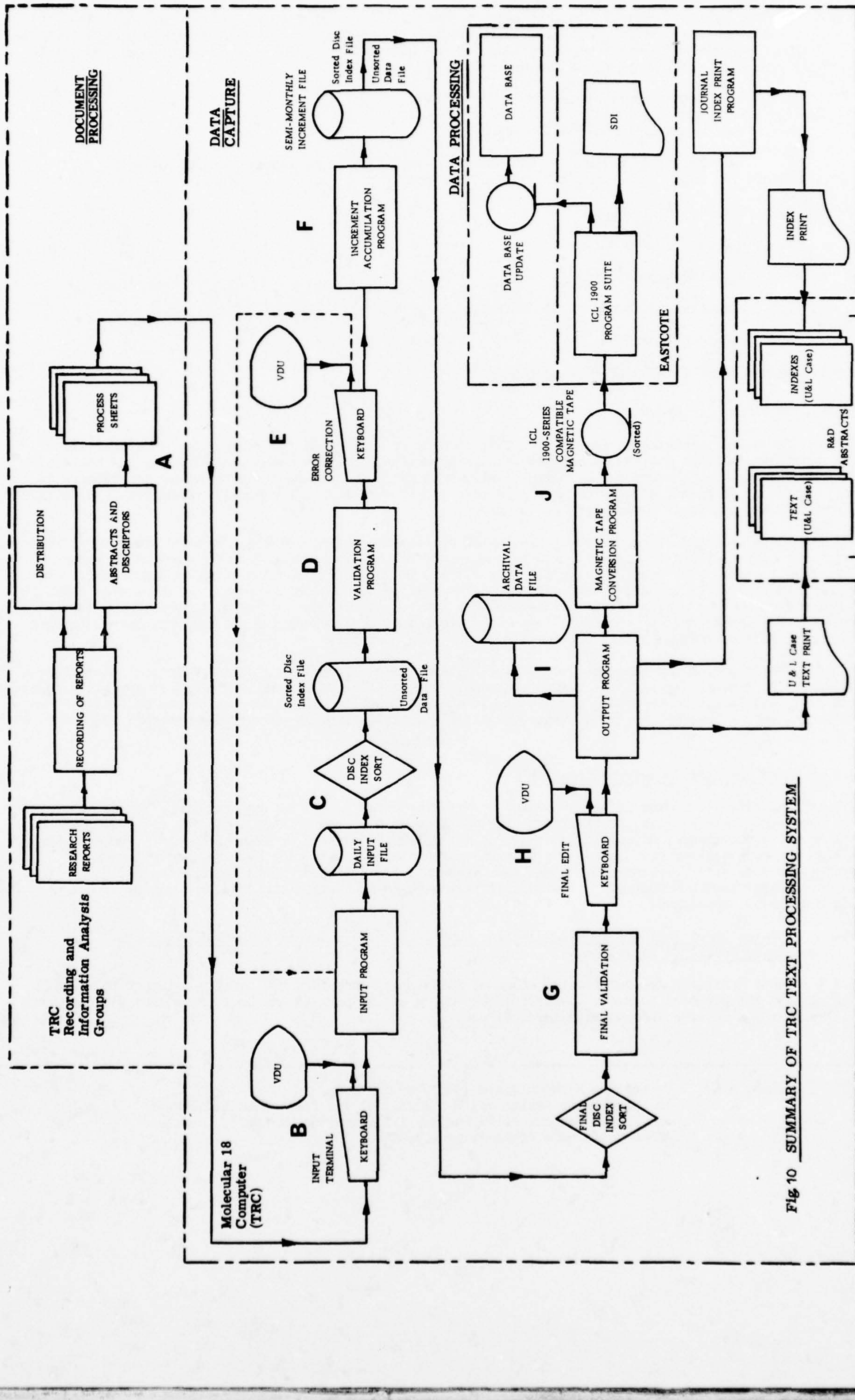


Fig 10 SUMMARY OF TRC TEXT PROCESSING SYSTEM

5. GENERAL DISCUSSION

5.1 The two systems described in this paper produce similar abstracts bulletins, yet there is little similarity between the systems apart from their use of mini-computers. I hope that these descriptions will suggest ways in which other organisations can use mini-computers for a similar purpose. Points to be considered when setting up such a system include:

a. is a computer necessary?

b. what computer (model and location)?

c. details to be included in the bulletin; should there be indexes to the bulletin; and should the documents be subject indexed (and how)?

d. form of input

e. processing procedures

f. output equipment

g. contents and format of the bulletin

h. source of software

I will discuss each of these briefly in turn.

5.2 Is a computer necessary?

5.2.1 The essential thing to remember is that a computer can not do the whole job. Thus analysis of the documents and data entry will need to be done by humans, and some human proof-reading and correction will also be needed. Moreover, the computer will need staff to operate it and possibly programmers as well. System design for a computer system is more complicated than for a wholly manual one, and decisions, once made, are generally more difficult to change.

5.2.2 The advantages of the computer lie in its ability to sort material quickly and accurately, to carry out simple checks on the data, and to enable corrections to be made without complete retyping. This last feature saves proof-reading time since there is no need to re-read a complete document. Two other advantages of a computer-based system are its ability to compile all manner of indexes with little extra effort and the availability of the same data for keeping records of the distribution and loan of reports, for invoicing, for recalling loans, for providing SDI services and for building up a data base for retrospective searching, not necessarily on the same computer.

5.2.3 Thus the pros and cons need to be investigated thoroughly before a decision to use a computer is made. In particular, to sell the concept to management, a detailed feasibility study is generally required, and even then it may be necessary to point to intangible benefits, which can not be costed, rather than to strict financial rewards, as the ultimate or partial justification. A useful paper in this respect is ref 7.

5.3 What computer (model and location)?

5.3.1 Choosing a computer is being covered by Mrs Grosch and I will say nothing about it here except to add that other possibilities apart from having one's own mini-computer and developing the system in-house are to use a mainframe computer at a bureau, or a time-sharing system or even to have a 'turnkey' type of system in which the computer is on your premises but another organisation develops and programs the system and runs it. There are advantages and disadvantages in each of these approaches, but having the computer under your own control - particularly in regard to scheduling runs and the quality of output - is often of major importance.

5.4 Details to be included in the Bulletin, should there be indexes to the Bulletin, and should documents be indexed?

5.4.1 These questions are outside the scope of this paper since they are equally relevant to manual systems and the proposal to use a computer should not play any part in the decisions. But the computer will enable you to have indexes with far less work.

---

7. Jones, A.C. Presenting a development plan for approval. In 'Government Assistance for Technical Information in Industry and Simple Mechanisation for Small Information Centres', AGARD Conference Proceedings 117, March 1973.

5.5 Form of input

5.5.1 There are many possible ways in which the data can be entered into the computer. The obvious ones are punched cards and paper tape but both these suffer from disadvantages. Both are extremely noisy and the electro-mechanical equipment tends to be less reliable than more modern electronic equipment. Moreover, both require the handling of discreet items, cards or reels of paper tape, which can cause problems if insufficient care is taken. Dropping a box of cards or tearing a paper-tape are both very time-wasting. Punched cards have the added disadvantage that the input can only be in upper case (at least I know of no upper and lower case card punches). Against these disadvantages must be set the fact that the equipment is fairly cheap to purchase.

5.5.2 A somewhat higher level of sophistication is obtained by the use of cassette or floppy disc encoders. These have the advantages of quietness and reliability; and handling a magnetic tape cassette or a floppy disc is far less likely to cause trouble than handling cards or paper tape. They usually provide facilities for formatted displays on a VDU and allow various validation checks to be made. If the mini computer does not have facilities for handling cassettes or floppy discs, a converter producing standard magnetic tape must be used and these are relatively expensive.

5.5.3 The next stage from stand-alone units of that kind is processor-controlled keying, where a group of such units is connected to a mini-computer supplied as original equipment with the data entry devices. Some systems of this kind are stated to be economic with as few as four 'stations', so they could be used for a medium scale operation. An added advantage here is the ability to retrieve records at will for verification, proof-checking or inspection by the data prep supervisor. In such a system, records can be kept of the number of items entered by each operator and of the number of errors (at least those thrown up by validation). Moreover, the validation checks can be more sophisticated than those available on a cassette or floppy disc encoder.

5.5.4 A variant of this, as used by TRC, is to enter the data directly to the mini computer via a VDU. The software controlling data entry and keeping the appropriate records has of course to be written in-house, but the advantage is that a second computer is not needed, making the system economic for one or two terminals. On the other hand no data input is possible if the computer goes down.

5.5.5 Optical character recognition is yet another approach. The usual method is to type the material accurately using a special typewriter with a suitable fount, eg OCRB, and then scan it with the reader. The cost of this approach is such that a small or medium sized bulletin would probably need to use a bureau rather than obtain a machine for in-house use. A more speculative possibility is to scan the appropriate parts of the report directly with a suitable reading device, perhaps hand-held, but I know of no suitable equipment. However, I imagine it will come eventually, as will voice input.

5.5.6 Coupled with data input is the question of checking the entries. Computer validation is generally only a limited check that fields are in the right format, that there are the correct number and that reference numbers and dates are in appropriate ranges. Fuller checking against the original document, particularly of fields such as reference numbers and authors which are likely to be used as identifiers or search keys, is desirable. This can be achieved either by proof-checking and correction as at DRIC and TRC or by verification (ie a second key-boarding operation). Verification of this kind is expensive and the need is questionable when dealing with running text such as an abstract where minor errors such as the omission of a space after a comma are of little consequence but would delay the operation. A visual check is probably best for abstracts. However, the added accuracy obtained by verifying using double key-boarding might make its use justifiable for other fields such as reference numbers, authors' names and subject index terms.

5.5.7 Subject index terms can be checked by eye or, if selected from an approved list or thesaurus, by computer validation against the list. This check is of most importance if the data are to be used as the basis of a batch retrieval service such as SDI where errors in spelling will mean that the document is effectively lost. For an on-line service with an 'expansion' command as in the Lockheed, SDC and ESA systems there is less need for an accurate check since variant spellings can usually be easily identified. However, incorrect use of partly or wholly synonymous terms such as wolfram and tungsten will cause difficulties even in an on-line system, so perhaps computer validation should be the aim.

5.5.8 In both the systems described in this paper, data input takes place only when the report has been completely processed. However, both DRIC and TRC are planning systems in which the basic record can be input to the computer shortly after the report is received. This approach, which has already been adopted by some other organisations such as NASA, means that queries about the report can be answered from the time of its receipt without the need for an additional manual record. However, system design is more complex and great care must be taken to ensure that the accession or other reference number is input correctly or there may be problems in reconciling the various parts of the record for each report. A check digit system may be useful in this context. A direct entry system is the best in such circumstances because staff entering details after the first stage can display and check the details already entered.

5.6 Processing procedures

5.6.1 The processing procedures used by DRIC and TRC have been described above. However, there are many other possibilities and the precise approach to be adopted depends upon the equipment and software available and the number of different outputs required. No guidelines can be laid down except to point out that computer sorting of large records is a lengthy procedure and so minimum-length records should be used at the sorting stage.

5.7 Output equipment

5.7.1 DRIC use an upper and lower case line printer at present, as do TRC for their indexes. However,

TRC use a golf-ball typewriter for the body of the bulletin to get a clearer product, and plan to use a daisy wheel printer for greater speed, and ultimately photo-typesetting. The last named requires careful thought since the equipment is expensive and might be economically justifiable only if a wide range of work can be found for it. However, it gives such extremely good results and allows so much flexibility in the size and style of type that it should be seriously considered if a suitable bureau is available and there are no problems of security. Using a bureau will of course add a further delay to the production of the bulletin. 'Daisy-wheel' printers, which are often found in word-processing systems operate at about 50 cps and may not be able to cope with the volume of material, particularly cumulative indexes. However, they are relatively cheap and so it might be possible to obtain more than one, provided the computer can service a number of them simultaneously.

5.8 Contents and format of the bulletin

5.8.1 Decisions have to be taken as to the items to be included in a bulletin entry and the exact format of the entry and what indexes if any are to be published. These are really decisions to be taken irrespective of the method of production of the bulletin, but they may help to influence decisions as to the size of computer and the peripherals, eg discs and printers, to be obtained.

5.8.2 Another aspect to be considered concerns page sizes and item formats. DRIC print both abstracts and indexes on continuous computer stationery at such a size that a 67% reduction gives an A4 page size for the bulletin. This is about the maximum advisable reduction ratio for ease of reading. It should be borne in mind, however, that paper is expensive and getting ever more scarce; and the greater the reduction, the fewer the number of pages needed.

5.8.3 The actual format is often a matter for compromise between aesthetics and computer capabilities. DRIC has recently adopted a 2-column lay-out for its indexes and TRC intend to do the same for the body of the bulletin but not the indexes when photo-typesetting is available. NTIS Government Report announcements are printed in a 3-column lay-out. These approaches stem from the belief that short lines of text are easier to read than long ones. A multi-column lay-out is possible with a normal line-by-line printer or typewriter but requires careful programming. For example the contents of the first column have to be written to magnetic tape or disc (preferably the latter) and called back during printing. However, some recent printers do allow several columns to be printed in sequence, the paper being re-wound at the end of each column except the last.

5.8.4 Some studies of the format of indexes have been carried out by the Readability of Print Unit of the Royal College of Art in London, although they do not comment on the length of lines. The most relevant study in this context is ref 8 in which different formats for an author index were studied. The most satisfactory format was one in which the whole entry apart from the author was indented by 2 character positions and there was no blank line between entries. A brief review of a presentation of the work of this Unit with a list of their publications is given in ref 9.

5.9 Source of software

5.9.1 The use of commercial software packages is being discussed by Mrs Grosch and so nothing need be said on that topic. It is of course, possible to place contracts with a commercial software house to write all the software. But if you do so, ensure that the specification is exactly what you want before the job is started, have a fixed price contract if at all possible, and check that the documentation is adequate. It is almost certain that the software will need to be changed after the system has been running some time and so good documentation is essential. The alternative of writing software in-house does ensure that last minute changes can easily be made, but the cost of setting up a programming team should not be under-estimated. Documentation is essential, even if programs are written in-house, to enable later changes to be made when the original programmer has left the installation or even if he has merely forgotten the details of what he wrote 3 years earlier.

5.10 SUMMARY

5.10.1 In this paper, I have tried to describe in some detail how one abstracts bulletin is produced by mini-computer and in less detail how another one is produced. There are of course many other abstracts bulletins prepared partially or wholly by mini-computer which I have not attempted to describe since I have no first-hand knowledge of them. Finally, I have summarised the various points that need to be considered, with a few comments on the pros and cons of various approaches.

ACKNOWLEDGEMENTS

I should like to acknowledge the help given me by Mr V Rogers of TRC and by my colleagues in DRIC, particularly Mr R H Howe, Mr J C Dunne and Mr A J Findlay, and I should like to thank the Ministry of Defence for allowing me to write this paper. Any views expressed in this paper are purely my own and do not necessarily represent the views of the Ministry of Defence or the Department of Industry.

---

8. Spencer, H. et al. The relative effectiveness of ten alternative systems of typographic coding in bibliographical material, Royal College of Art, London, 1973.

9. Hart, G.W. Readability of Print Research Unit (Report of a Seminar organised by the Unit, 22 April 1977) DRIC-MEMO-2-77, Aug. 1977.

REFERENCES

1.	Williams, P.W. and Curtis, J.M.	The Use of on-line information retrieval services, Program Vol 11 No 1 Jan 1977 pp 1-9
2.	Johnston, Susan M, and Gray, D.E.	Comparison of manual and on-line retrospective searching for agricultural subjects, Aslib Proceedings, Vol 29 No 7 July 1977 pp 253-258
3.	Hart, G.W.	The use of a mini-computer at the Defence Research Information Centre. In 'Advancements in Retrieval Technology as related to Information Systems'. AGARD CP-207, October 1976.
4.	Ridler, E.H. et al.	Production of R&D Abstracts and printed indexes to R&D Abstracts using tape typewriters. TIL report 21, Aug. 1969.
5.	Defense Documentation Center.	COSATI Subject Category List (DOD Extended), AD 624000, December 1965.
6.	Adams, H.C.	Computer processing of bibliographic data at the Technology Reports Centre of the Department of Industry, TRC, 1977.
7.	Jones, A.C.	Presenting a development plan for approval. In 'Government Assistance for Technical Information in Industry and Simple Mechanisation for Small Information Centres', AGARD Conference Proceedings 117, March 1973.
8.	Spencer, H. et al.	The relative effectiveness of ten alternative systems of typographic coding in bibliographical material, Royal College of Art, London, 1973.
9.	Hart, G.W.	Readability of Print Research Unit (Report of a Seminar organised by the Unit, 22 April 1977) DRIC-MEMO-2-77, Aug. 1977.

Reports quoted are not necessarily available to members of the public or to commercial organisations

SELECTIVE DISSEMINATION OF INFORMATION

By
R.A. McIvor
Director Scientific Information Services
National Defence Headquarters
Ottawa, Ontario
K1A OK2

SUMMARY

A Selective Dissemination of Information (SDI) system is described which has been implemented on a minicomputer. The first part of the paper discusses the preparation of profiles, and the role of the information scientist in their testing and maintenance. The second part of the paper discusses the methods of implementation of SDI programmes, in particular the trade-off between memory capacity and speed which must be faced. The current matching algorithm at DSIS is described in greater detail. Somewhat over 1,000 questions are matched every two weeks against about 2,500 document records, producing about 12,000 individual retrievals.

One of the more popular applications which automated information systems have facilitated is the Selective Dissemination of Information or SDI service. Although this service existed in our centre and in many others before automation by sending copies of catalogue cards of recent accessions to users based on either personal or recorded knowledge of their interests, the introduction of automation has greatly increased the scope of such services. From a small beginning in 1969 with some 25 user questions surveying some 250-300 acquisitions every two weeks, our service has grown to some 1,100 questions covering close to 2,500 documents every two weeks, and in addition an SDI service is provided on project reports of research in progress but not yet formally reported.

Although the programs have changed several times over the years, the basic method of inputting a question into our system has altered little (Figs 1 and 2). Although details of entering questions into an SDI system vary extensively from system to system, the general principles are the same. The method used at DSIS is similar to one developed at Chemical Abstracts for searching their Chemical and Biological Activities tapes.

Firstly some form of question identifier is needed. This is used, amongst other things to record the "question weight" when this feature is permitted. In our system it also provides a preliminary screening of the items to separate those which this user is entitled to see and the user's name and address. This preliminary screening serves as a safety check in controlling the security or sensitivity level of the document retrieved. Following this are a series of term identifiers. Terms are grouped into what we call "parameters". Terms in a given parameter are connected by "OR" logic--that is, the occurrence of any of these terms in the appropriate place will count towards a retrieval. However, parameters are connected by "AND" logic i.e. at least one term from each parameter must be satisfied to trigger retrieval. In our system terms can also be identified as "NOT" terms. Any satisfaction of a NOT term prevents retrieval of the item. NOT terms are used amongst other things as a further limitation on security and sensitivity tailored to a particular user need-to-know. Another feature of our system is weighting. Terms may be given a term weight, and the sum of satisfied term weights must equal or exceed the question weight mentioned earlier if the item is to be retrieved. Weights can also be negative if desired.

Other items included with the term restrict its scope to a particular field or fields - for example to author, corporate source or subject index; indicate the mode of search and whether it is an OR or NOT term. The mode can be prefix, suffix, infix or full word or phrase. Individual terms are often truncated to decrease the number of terms required, and thus increase the number of questions that can be searched in one pass.

We have found in practice that few people are willing to spend the time and effort to construct or modify their own profiles, and when they do, the results are much less satisfactory than when done by a professional. We have a number of information scientists on staff with different subject specialties. Before automation, their chief tasks were the indexing and abstracting of reports coming into the system and offering some literature searches on request, and operating a rudimentary manual SDI system. Over the years, the intake of documents requiring indexing and abstracting has reduced to one-quarter what it was, since the largest proportion of incoming material is in the form of microfiche already indexed and catalogued in machine readable form and requiring only the conversion of the magnetic tapes to our format. The information scientist now spends a great deal of his time preparing and modifying SDI profiles for our users, and providing custom retrospective searches on our own and commercial data bases. He thus can become very familiar with the different vocabularies used by different sources to adapt the search profile to the different source materials. Generally, the user describes his interests to the information scientist who prepares a trial profile. We have a facility for on-line testing of the trial profiles, but more usually it is included in a regular run and monitored for several weeks by the information scientist until he is satisfied that he has achieved a proper balance of recall and relevance to satisfy the user. We have found that users differ considerably in their tolerance of irrelevant material. Some users prefer to be sure of almost total recall at the expense of considerable irrelevant material while others are annoyed by even a few irrelevant items.

Once the information scientist (I.S.) is satisfied with the profile, the retrievals are sent automatically to the user; but the I.S. continues to monitor the statistics of retrievals to watch for any profiles receiving either excessive or few or no retrievals. He should also keep in regular contact with his users to ensure the material received is satisfactory, since users are often more inclined to complain

about inadequate service or have it cancelled completely than to make an effort to have it improved. A number of customers often obtain no hits because of narrowly defined subject areas. Others are quite content to sort through up to 80 hits. An average profile runs between 0-25 hits per 1,000 documents searched.

Although in principle any field can be searched, in practice we no longer allow scanning of the text of the abstract. It was found by early experience that precision (i.e. relevance of the retrieved material) when abstract searching was permitted was very much reduced without a large increase in recall, and such searching is considerably more costly than searching of index terms. We have found the best type of profile construction is to have a parameter of COSATI subject fields to limit the range of subject matter searched and one or more parameters of index words or word fragments.

While we have discouraged abstract searching, we now arrange for multiple field searching when the index-term field is specified. Our index-terms are divided into three fields - one contains only those terms from the Thesaurus of Engineering and Scientific Terms (TEST), at least for our own records, and a second contains words from our own controlled vocabulary which covers specific item names or concepts not covered adequately by TEST. The third contains broader terms from the TEST hierarchy. For example if the term - Saponification were inserted by an information scientist, the progressively broader terms Hydrolysis, Solvolysis, Decomposition reactions and Chemical reactions would be automatically added by computer to this third field. Items from this field are not included in our Document Digest indexes, but are very useful for Boolean (combination) searches in SDI. The final field automatically included is the document title. If all our records had been catalogued in DSIS this would perhaps not be necessary, but some of our data bases do not use the same controlled vocabulary as ourselves, and the use of the title brings about some retrievals that might otherwise have been missed.

Weighting of terms can be another means of realizing greater precision in the SDI output. However, it must be used with caution. In general, it is more difficult to construct a satisfactory profile with weights, than when Boolean logic is employed. Depending on the search method used, multiple occurrences of the term in the same field (or set of fields being searched as one unit) may lead to multiplication of the weight by the number of occurrences. If this feature is present the weighting of general words like detection should be avoided.

The retrieval form (Fig. 3) sent to the customer has a very flexible format which can be readily altered for different data bases or for special requirements. The usual format contains all material necessary for a library to identify a document plus an abstract (if available) and the index terms. We also add an order block making it very simple for the customer to obtain the document. We do not necessarily have copies of all announced documents of foreign origin. To save effort in such cases we use the same form to order the document through our foreign liaison staffs.

There are a number of ways in which an SDI program can be implemented. While the method chosen does not affect the user in the broad outlines indicated earlier, there will be some differences in detail.

Two general patterns of implementation are prevalent. One is frequently used when the number of users is much larger than the number of documents to be screened for a particular issue. It consists of forming inverted files (i.e. indexes) of the document file to be searched for each field to be searched, and matching the index one-by-one against the search profile to determine which documents match. This technique can be quite efficient, and, if the retrospective on-line search facility is offered the resultant file can then be used for updating that system. It has the disadvantage of not being readily adaptable to infix and suffix searching, and the preprocessing may require more storage and memory requirements than are available in a small system. We have not used this method at DSIS.

The second approach is to read the documents sequentially and compare them against the profiles either sequentially or after preprocessing. This method is quite adaptable to small systems and all the features I have described, and is the system that has been used at DSIS. Our first system used only 8,000 words of 16-bit memory and could search 25 questions at once. It was however much too slow for consideration when the number of questions ran into the hundreds. In particular, profile terms were stored two characters per 16-bit word, and the minicomputer we possessed had no instructions that could deal with these two halves, called bytes, separately. When we acquired more memory we stored one character per word in both the record field to be compared and the profile terms. Both were converted to all lower-case for comparison and all punctuation marks were converted to blanks. Using this technique with 16K of memory, 50 profiles could be searched in each pass through the search file.

With further refinements giving another 4-fold increase in speed, this programme served until a year or so ago, despite the fact that it was becoming more and more unwieldy. With over 1,000 questions to be matched against some 2,000 documents, some 21 passes were required at about 40 min per pass or a total of some 14 hrs computing time per search. Although this was run unattended during the night, any momentary power cut, as well as a number of other factors could bring the search to a halt or invalidate the results. With a single computer and its peripherals, the number of retrievals that could be sorted in a single step was limited. This number was being more and more frequently exceeded, and the use of both computers working together was then necessary. Although the sort phase was only about an hour, this too had to be done in non-working hours because the other computer was dedicated to data input during the day.

With the upgrading of the CPU and the availability of additional memory a new algorithm could be considered. This used two sixteen-bit words for each character in the search term, however the search terms were built into tree structures, so common strings of characters needed only to be searched once per pass. In particular since there are at most 36 characters that can begin a term, only 36 comparisons are necessary for the first level of as many questions as can be compiled together. This number is variable, as all available memory is allocated during the profile compilation phase. In practice, about 150 questions are searched simultaneously. It was also possible with more disc space to keep a copy of all retrievals on disc. The file of hits could then be sorted and stored after every pass, eliminating a separate sort step. With these modifications, an entire search beginning with profile compilation to the preparation of a final

print tape takes about 90 minutes - approximately 30 for the compilation and search and 60 for the preparation of the print tape. Such a search would produce about 12,000 matches for printing off-line. The printing is done with a Xerox 1200 which prints one page/sec, thus requiring about 4 hours for printing. The times are now such that this can be easily done during regular hours.

Since this algorithm change has made such a dramatic improvement in search time, it might be worthwhile to describe it in more detail. Each character is alloted a 16-bit word for ease in searching and a second word as a pointer to alternative strings. Let us assume we have the following terms with the users for the purpose of the example indicated by an upper case letter in brackets: cattle (A), dogs (A), donkeys (A), cat (A), catalysis (B), catalytic (B). These would be stored as in Fig. 4.

FIGURE 4

ADDRESS	LETTER	POINTER	ADDRESS	LETTER	POINTER
1	C	8	19	(A)	20
2	A	0	20	L	0
3	T	19	21	Y	0
4	T	0	22	S	26
5	L	0	23	I	0
6	E	0	24	S	0
7	(A)	0	25	(B)	0
8	D	0	26	T	0
9	O	0	27	I	0
10	G	13	28	C	0
11	S	0	29	(B)	0
12	(A)	0			
13	N	0			
14	K	0			
15	E	0			
16	Y	0			
17	S	0			
18	(A)	0			

A pointer of 0 indicates no further string occurs. An upper case letter in brackets indicates a hit for the indicated user. Let us see how this works. Assume the string "CHOW DOGS ARE" occurs in the text. Matching begins at address 1 with the C of CHOW. There is a match so the next letter of CHOW, which is H, is matched against the address 2 which contains A. As the match failed, the pointer at 2 is examined for a continuation. There is none, so matching beginning with the C of CHOW is abandoned. Now matching begins again at address 1 with the H of CHOW. There is no match, and the pointer at C is non-zero (8) so a new match is tried at address 8. H does not match with D and there is no continuation pointer so the search on H fails. Similarly, the 'O', 'W' and blank characters fail to match. Now 'D' is tried and fails to match at address 1 but matches at the continuation 8, matching continues at 9 with O, 10 with G, and 11 with S. When 12 is accessed to match with blank, it is discovered to be a retrieval flag and a hit is recorded for user A. There is no further pointer at address 12 where other hits or a character string continuation would be indicated, so the matching restarts at address 1 with blank etc. No further hits are found in this example.

The retrieval form (Fig. 3) has been rearranged to improve its appearances, at the expense of processing time, and an order form is printed on the sheet in the same printing operation, using another facility of the Xerox 1200. The use of this form has reduced much of the paperwork in dealing with requests.

From a small beginning in 1969 our SDI service has grown to be our most important service, and the one which gives us the most direct contact with our users. The service, as provided originally, was run on an IBM 360/65 using programmes written in the FORTRAN language. With the speeds and costs encountered during this phase, using the programmes then in vogue, it would never have been feasible to expand the service to individual users as we have done. While the structure presently used on our minicomputer would have probably made the use of the programme feasible on the larger system, I hope to have shown that the minicomputer can do the job, and in many cases, is a much more desirable solution.

Cost-effectiveness in Library Automation

J.H. Ashford, BSc. Ph.D. FGS
Divisional Manager, Lipman Management Resources Limited
54-70 Moorbridge Road, Maidenhead, England.

Summary

The advent of relatively low-cost minicomputer systems has created opportunities for cost effective library automation. In trying to achieve best results, a number of factors characteristic of library needs have to be taken into account. These include the interleaving of high volume and high complexity processes; the need to manage textual material where the meaning and use of the content are context dependent; and the requirement to install systems which are easy to use and robust for 'non-computing' users.

Containment of development cost appears to be most readily achieved by collaborative design - where a group of libraries come together to share design resources and costs - and by the careful re-use of previously developed systems. National and regional libraries can make major contributions in bibliographic systems and a range of co-operative and other commercial services are proving themselves. It is proposed that the formation of shared development groups enables librarians and computer staff to maintain continuity of experience and growth of expertise both in the 'librarianship' aspects and in minicomputer system technology.

Context

During the last decade, library automation has moved from tentative experiments to a well established, although notoriously difficult, application area. The advent of the 'mini-computer' offers the chance to develop library dedicated systems on an economic basis and to move away from the earlier constraints of sharing equipment and software resources with other projects.

This is a matter of some importance, because library systems differ from the general computer application in several aspects:

a) Straightforward but high volume operations (as in control of issues) are interleaved with complex, low-volume work (as in cataloguing).

b) Data files are normally large, often slow moving. (40 to 2000 million characters of on-line storage is the current range; 200 million characters is typical)

c) The use of much of the data on file, especially in information work, is context dependent and involves a degree of 'meaning' to be derived from textual matter.

d) The system user, as librarian or researcher, is often not skilled in computing methods, and may have little numerical or technical training.

- and, perhaps especially, the meaning and value of the data are dependent on the expectations and experience of the user, and so library systems can be extended and complicated without apparent limit, and certainly beyond the ability of the systems engineer to comprehend them and describe their functions in the exact and predicative form required for computer implementation.

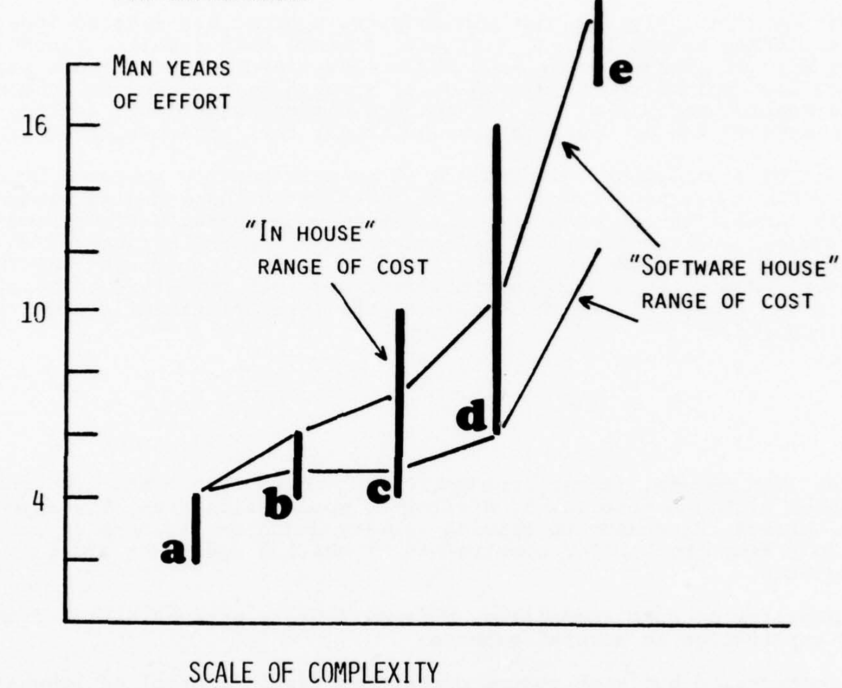
The attributes of minicomputers are well described elsewhere. For the purposes of this paper, the following salient characteristics are noted:

a) Hardware - both computer and storage - is cheap, and it is normal to add to machine capacity rather than complicate programming.

b) The mini-computer standard software, though often excellent, is generally circumscribed, and the systems designer requires a much more detailed knowledge of operating system and related hardware than in a mainframe context.

c) Many mini-computer manufacturers prefer to sell wholly or mainly to systems engineering companies who build systems incorporating the mini for the end user.

FIGURE 1. COMPLEXITY / COST DIAGRAM FOR CATALOGUING



a SIMPLE BATCH APPROACH, NON-MARC

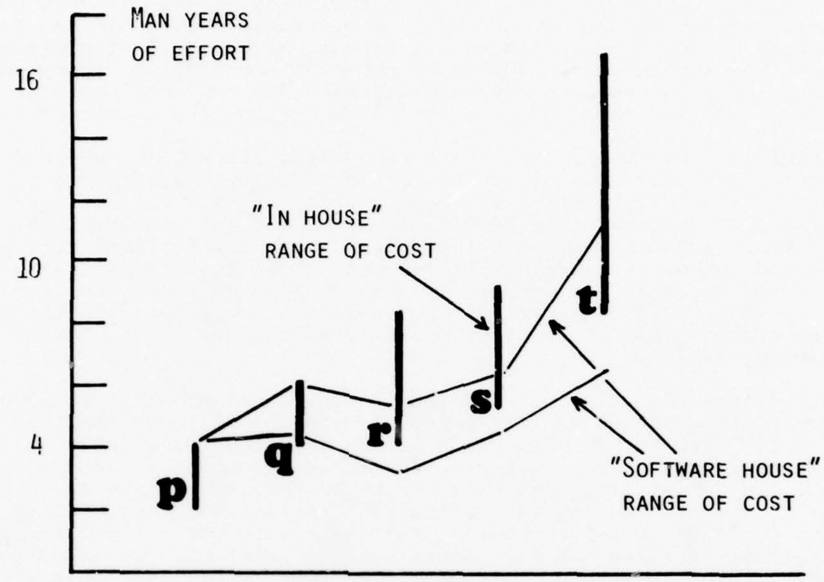
b SIMPLE BATCH WITH MARC INPUT

c ON-LINE, SIMPLE, SINGLE USER

d ON-LINE, SIMPLE, MULTI-USER

e COMPLEX MULTI-USER, LARGE NETWORK

FIGURE 2. COMPLEXITY / COST DIAGRAM
FOR CIRCULATION



p SIMPLE BATCH: CHARGING, DUES, STATISTICS

q SIMPLE BATCH WITH RESERVATIONS

r SIMPLE ON-LINE

s SIMPLE ON-LINE WITH RESERVATIONS

t MULTI-SITE, ON-LINE

Scale and origin of development costs

Figures 1 and 2 use data from a variety of sources to indicate the range of software development costs experienced in library automation projects over the last few years. The majority of the cases used were designs from first principles; mini-computer and mainframe schemes did not differ significantly in software cost. The costs are expressed in man-years so that the comparison can be made between the effectiveness of in-house and good quality software company teams; the latter gain substantially from their concentration of experience and high calibre staff, but their fee per man-year is likely to run at two to three times the cost of in-house staff.

Major costs arise in the following areas:

a) For circulation control: Costs arise in the large volumes of data to be handled and the necessity for high accuracy when the eventual 'lost book' has to be attributed to a (usually indignant) delinquent borrower. As the cost of books and periodicals increases and the rules on copying become more restrictive, the security element of circulation control becomes more significant, and the complexity of software necessary to achieve very low rates of error and high credibility increases rapidly.

b) In information retrieval: There is a recognised and well documented 'trade-off' between the effectiveness of an information retrieval system and the computer resources consumed in running it. Less obvious is that the software complexity of many established IR systems stems from their batch mode of operation and that a determined design effort is necessary to get accepted the relative simplicity possible on on-line systems and not carry over interesting but costly features of batch working. There is also a major increase in software cost in going from controlled language (thesaurus-based) systems to 'free language' or 'whole-text' working although in the views of many researchers the latter offer operational savings. (See, for extended treatment, Barraclough, 1977.)

c) In cataloguing: The principal problems in the development of cataloguing systems seem to stem from the necessary complexity of managing whole text either for input or output handling, for forming and using access keys, or for sequencing files on extended text fields in the preparation of 'alphabetic' sequences. One can consider whether or not to implement diacritic marks (easy in English, difficult in French) or even whether to maintain UK-English/US-English distinctions, but the complexities of genitive prefixes (M'Kay, McAndrew, MacDonald, and for confusion Mack and Machinery) and of gender suffixes (chien/chienne but not chiot/chiotte) tend to surprise systems analysts and programmers not previously exposed to bibliographic work.

Now, most of these problem areas in development are well understood - at least somewhere - and the principles, though rarely the details of successful practices, have been written about. The Co-operation in Library Automation study (Ashford, et al. 1974) investigated a number of cases of collaborative development, and while the conclusions on the scale of such co-operation at that time were a disappointment to the authors, they concluded that savings in the region of 25% to 75% of the investment cost of an automation project could be made by re-use of existing system designs or entire systems. Furthermore, in several cases where libraries had considered and rejected the use of a pre-existing system as unsuitable, the comment was made that the time spent in review and trials was not lost, both because of the experience gained and also because of the sharpening up of the objectives of the design team.

Cost reduction and containment

Perhaps the most effective way of reducing system design and programming costs is to avoid writing unnecessary programs! This involves careful and detailed specification of the requirements of the users of a proposed system in the language of the user, re-expression of this functional requirement as a system design, using the language of the computer systems analyst, and then thorough checking of every segment before it is accepted for development for unnecessary elaboration, re-development of existing work, failure to make use of available files, and, on the positive side, for giving the user a service at least as flexible and robust as the overall library system of which it will form part.

Key questions are:

Circulation control:

Is this a system for 'informing on the location of material' - or is it 'protecting property'?

Are 'reservations' a necessary feature?

Do the 'statistics' to be collected on the operation of the system relate to real management issues, or are they just interesting data?

Do we need a computer based system at all?

Information retrieval:

Why do we need our own system at all? (i.e. Is our material so secret, special, obscure, voluminous that we cannot incorporate it in a generally accessible on-line service?)
Will the retrievable entries be regularly accessed over a long period - which indicates indexing and controlled language - or does the file contain infrequent use material and ephemera, suggesting whole-text searching?
Must whole documents be stored on-line? - or will abstracts only suffice? - or can we run an adequate service on citations only, with abstracts on, say, indexed microfilm?
Does the 'user interface' have to be suitable for trained staff, or must it be practical for the intelligent but untrained user?

Cataloguing:

Is the catalogue going to be primarily a retrieval tool (short entry) or is it also going to serve bibliographic functions?
Can the entries be drawn from a national or other bibliographic centre - either entire, or with local modification?
Is there a substantial volume of 'local' entry - or can the specials be done using an on-line bibliographic service?
How do the planned, true, average costs per entry compare with Birmingham Libraries Co-operative Mechanisation Project, for example, at less than £0.70 per entry for 15,000 titles per year including Computer Output Microform or card output? (Other bibliographic services are comparable)
Does the user (as opposed to the cataloguer) need on-line access to the local catalogue, or is a printed or Computer Output Microform catalogue more practical?
How many access points are required - author (fairly easy); title (easy); combined author title (fairly easy); class number (fairly easy); subject (can be awkward)?

Management:

Does the library have the skills available for development (and later for operation) of an in-house computer system?

In the cost justification for the project, have we used true costs, or is the costing affected by budgetary rules or administrative quirks which are arbitrarily variable to the detriment of the project economics?

Is there any possibility that the project is being undertaken for 'research' purposes, or even for reasons of personal or departmental prestige? - because if this is the case, the above test questions become very difficult to answer objectively!

What positive steps can be taken to minimise costs and still achieve a worthwhile and effective system?
Firstly, one can use the experience of those who have recently installed systems, and often, because librarians belong to a communicative and generous discipline, have free access to much detail of their work. Seek design knowhow, re-usable software, re-usable manuals and presentation material, and for files of bibliographic or index material on computer media. Sources are Aslib Information, Journal of Library Automation, Library and Information Service Abstracts, Program, VINE (See bibliography). Secondly, it is usually profitable to review the products and services offered commercially, particularly in cataloguing and information retrieval - in Europe, Automated Library Systems Limited, British Library, Blackwells, Birmingham Libraries Co-operative Mechanisation Project, Lipman Management Resources Limited, Plessey, among others, are in various ways engaged in providing automation services to libraries.

At this stage one may have found most or all of a suitable system; often, however, local requirements and specialisation push the project towards new implementations. The next step is to consider whether partners can be found for a collaborative development to spread the capital costs over several institutions. The history of such group projects in library automation is good - Ohio College Library Center was one of the earliest, Birmingham Libraries Co-operative Mechanisation Project and South West Academic Libraries Co-operative Automation Project are well established in United Kingdom, and Scottish Libraries Co-operative Automation Project is doing well at an earlier stage. It has proved practical to handle quite diverse libraries within each system - Birmingham Libraries Co-operative Mechanisation Project for example started with two academic and one public library and has now over eighteen members in UK and Western Europe.

The planning and specification stage of a collaborative project seems to take two to three times as long as for a 'single user' system; however, given three or more participants, not only do the overall shared costs come out beneficially, but the exactness of specification necessary to satisfy the collaborating partners before implementation results in a very low (and satisfactory) level of post installation amendment. The Birmingham Libraries Co-operative Mechanisation Project Final Report (1976) is highly informative.

If it is not practical to share one's development costs by collaboration, it is still often possible to keep control of some major risk areas by subscribing to established services. For instance, both the British Library and Birmingham Libraries Co-operative Mechanisation Project offer in the United Kingdom, automated bibliographic services of high quality and wide coverage. Within a few years, the scope of these bureaux services will be so wide (say, British National Bibliography since 1950; Library of Congress since 1972; 30,000 or more serials records and more than 100,000 other 'non-MARC' records) that careful selection of one or more sources for this 'generally used' range of material, should effectively eliminate cataloguing of non-specialised material for the individual library. Special skills and resources can then be applied to the analysis and indexing of <u>content</u> of material held in the library which more often does require local knowhow.

A final point from personal experience, is that mixed discipline development teams, with both librarians and systems analysts participating in all aspects of the work, get best results and build fewest misconceptions. This approach also helps to provide better continuity of technical support within the library once the initial development is over, since the library staff involved in the team tend to be less mobile than the traditionally volatile computer specialists.

Bibliography:

1. Program. 1966-
2. Journal of library automation. 1968-
3. SCOLCAP Newsletter. 1976-
4. VINE: A very informal newsletter on library automation. (The Library, University of Southampton, SO9 5NH)
5. Lancaster, W. ed. Proceedings of the clinic on library applications of data processing. (University of Illinois, 1972-.)
6. Ashford, J. et al. Co-operation in library automation: the COLA project. (London: LASER, 1975. 75pp.)
7. Barraclough, E.D. On-line searching in information retrieval; Journal of Documentation, vol.33 no.3 p.220-238. (London, Sept. 1977)
8. BLCMP Final report. (British Library Research & Development Department Project No. S1/G/027, 1976)

COMMERCIAL DATA BASE MANAGEMENT SYSTEM (DBMS) SOFTWARE

IN

LARGER MINICOMPUTER CONFIGURATIONS

Audrey N. Grosch
Associate Professor / Special Projects Officer
University of Minnesota Libraries
Minneapolis, MN 55455 U.S.A.

SUMMARY

Presently, the larger manufacturers of minicomputers in the United States offer commercial data base management system (DBMS) software for use in their medium to large scale minicomputer configurations. Most of these products are versions of DBMS which have been successfully in operation on large main frame conventional computers for nearly a decade. Comparison of DBMS to data management system (DMS) software and a brief historical overview are presented as a background to a discussion of DBMS and the design of online systems for libraries. Some questions are posed to help a given library determine whether it should use a DBMS approach in its online systems whether within a minicomputer or conventional computing environment. Specific minicomputer DBMS discussed are Hewlett-Packard's IMAGE/QUERY 3000; TOTAL as implemented on Digital Equipment Corporation PDP-11 Series minicomputers; DBMS-11 for the PDP-11/45 and PDP-11/70; and MUMPS-11 for the PDP-11 Series. We also present a discussion of the data structures supported, the language facilities, the minimal hardware configurations and various other component features as well as a comparison among these systems from the standpoint of potential bibliographic systems use. A brief comment on the importance of the data administrator function in a successful DBMS implementation is mentioned in the concluding remarks.

INTRODUCTION

In the last ten years commercial data base management system (DBMS) software has been developed and rather widely applied in large-scale computer systems -- particularly in management information systems applications or other largely non-computational computer applications. The development of DBMS actually began in the early 1960's when a number of different investigators determined that it would be desirable to separate data definitions from application programs and use the concept of generalized storage structures for data. This would permit multiple use of a single storage structure by different application programs. With this brief scene setting, let us look at what is happening in the DBMS area relative to the minicomputer environment as opposed to the traditional main frame computer environment.

DBMS DEFINED

Before any further discussion, one must prevent any confusion between the class of software called data management systems (DMS) and DBMS. DBMS software maintains and manages data in a non-redundant structure for the purpose of being used or processed by multiple applications. It organizes elements of data in some predefined structure and uses certain techniques to retain relationships between these different elements of data where such relationships exist and are critical to the applications being served. Application programs in this environment need only refer to the data logically by data item name, usually translated through a control module accessing a subschema object module.

On the other hand, a DMS software capability permits user programs to access, retrieve, and copy from a file -- usually predefined for a specific single application. A DMS may have facilities to minimize data redundancy and centralize storage of the data. However, its principal intent is to perform the functions of data retrieval, report generation and inquiry for a single application. Application programs in this environment must know the physical relationships of logical records in a file.

DBMS SOFTWARE OVERVIEW

Development of DBMS software in the minicomputer environment has somewhat paralleled that of software in the conventional computer field. Because of the generalized nature of DBMS and the desirability of supporting multiple types of data and file organizations, only the larger minicomputer system configurations can offer anything close to the capabilities of systems implemented within the conventional computer as a host hardware system. Application use of DBMS in a minicomputer system within a library would, in most cases, involve complex data item relationships, large file sizes, and multiple terminals performing

both data entry and retrieval applications. Thus, our view of DBMS in this paper will be restricted to those systems, in this author's opinion, which have the capability to perform those functions required if library management systems were to be designed in the DBMS environment rather than in the traditional single application approach.

Most of the commercial offerings of DBMS in the minicomputer field are systems which were originally developed for large-scale computers. Their degree of generalization requires a relatively large memory allocation. The memory requirements for these systems, at the minimum will require at least a 64K byte capacity to hold the operating system and a DBMS but may require up to 512K bytes for the operating system and DBMS in a multi-terminal, multi-user operating environment. Using Digital Equipment Corporation PDP-11 Series minicomputers as a representative example of the class of system required, a PDP-11/34 with 32K words and two 86 million byte disk units would be a minimal configuration. At the other end, a very large configuration would require a PDP-11/45 or PDP-11/70 operating under the IAS operating system with from 256K-512K words of memory with at least two 176 million byte disk units and other peripherals customary to a system of this size.

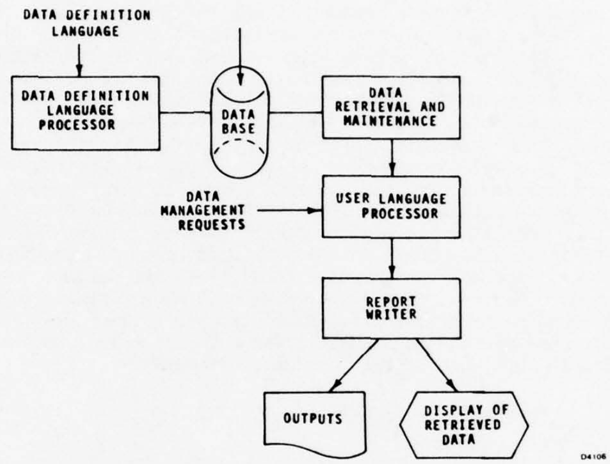


Figure 1. DBMS Functional Components - a Generalized View Across the Systems Currently Available

Figure 1. shows the functional components of a DBMS from a generalized view. From these modules stem the following general capabilities:

- Application program independence from the DBMS control program modules,
- Support of one or more higher level programming languages to be used to code application programs processing the data base content,
- Maintenance of user data definitions, i.e. logical data relationships,
- Mapping of logical data onto the physical storage devices by developing a data organization scheme,
- Utility programs which facilitate creation and maintenance of the data base,
- Data reorganization facilities,
- Data security and data access safeguards,
- System failure and recovery capabilities, usually through an automatic restart module, and
- System facilities for "fine tuning" of the DBMS physical structure according to experiences within the particular user environment encountered after initial installation, including performance vs. storage trade-offs.

In the minicomputer system, the smaller hardware system versions offer the basic capabilities above, but usually support smaller sized data bases and do not have the "fine tuning" options. These subset versions are normally upward compatible to the larger versions. Some of these have been optimized to maximize the inquiry/response time performance at the expense of increased mass storage utilization.

LIBRARIES AND DBMS

Currently, there are two approaches possible for libraries wishing to design library management systems or bibliographic retrieval systems utilizing the DBMS philosophy. The first approach would be to design and implement a DBMS optimized toward the applications common to libraries such as bibliographic data entry, fund accounting, indexing, inventory control, in-process functions such as ordering or claiming, and perhaps some specialized application as serials management. Work in this vein has been undertaken starting in late 1972 in one library in the United States, but is still in the late stages of advanced development. In other words, a considerable investment in programming is needed to undertake such a development which, unless an exportable product results, would be a heavy burden from a cost viewpoint for most if not all libraries. Clearly, this first alternative is not a practical one for any but the largest and wealthiest of libraries to consider.

The second alternative is to utilize an off-the-shelf generalized DBMS and build any necessary additional functions through application programs written in a supported higher level or assembly level language for the particular minicomputer. Depending upon minicomputer and specific model, other commercially available software can materially assist in the development effort. For example, inquiry/retrieval and report generator packages are available from most manufacturers for their systems capable of hosting a DBMS package. Cathode-ray tube screen formatting utilities are also available or can be created relatively easily to assist in the application input/output screen definition, testing, and installation.

In taking the second alternative, the user will require a shorter time to develop a working system, less manpower to create, install and support his system, and will maximize his personnel resources toward problem-oriented analysis rather than detailed coding of system software type functions.

Do libraries need to use DBMS in the development of bibliographic systems? To answer that question it might help to first consider answers to the following six questions:

[1] Does dispersion of data in scattered files lead to large amounts of redundant data with inconsistencies?
[2] Does this redundancy present problems in keeping the data current and making correct updates?
[3] Does data complexity lead to long lead-times for new application development?
[4] Do requirements for new data items or data relationships require and cause extensive application program modification?
[5] Is useful data so scattered in location, both in manual or different computer systems, such that it is unavailable for use with new applications?
[6] Is access to data primarily random, or non-sequential?

In the case of most library and bibliographic applications, the answer to each of the above questions is usually "Yes". If this is the case for your particular library, then you would be improving your system development effort by choosing an appropriate minicomputer configuration hosting a carefully chosen operating system and DBMS included in its software repertoire.

SPECIFIC DBMS ADAPTABLE FOR BIBLIOGRAPHIC AND LIBRARY SYSTEM USE

In selecting a minicomputer system for library use, the available software will influence system choice quite heavily if a commercial DBMS capability is to be chosen. Most of the larger minicomputer manufacturers offer some degree of DBMS capability in their larger systems -- many through special licensing arrangements with software vendors who have initially developed earlier versions of their DBMS for large main frame computers; some through supplying availability information on software developed and supported by third party software vendors.

Our view here will be largely restricted to those proven DBMS products available in 1977 for several of the larger minicomputers from the more well-known manufacturers and attractive for library and information retrieval use. In the next several years more software of this type should be available which, in this author's opinion, will have a tendency to be optimized for certain kinds of application environments. Hopefully, this will improve the user's ability to create systems of improved responsiveness with lower development costs.

To effect some comparison of the DBMS software to be discussed here we will consider:

- Data structures available, i.e. relationships and schema,
- Tools necessary for creation and use of the data base, i.e. data description language and its interpretive routines,
- Operating system required for the particular DBMS,
- Minimal hardware configuration required,
- Data security and access protection provisions,
- System restart and recovery provisions,
- Report generation and output facilities, and
- Input auditing and error handling facilities.

The specific DBMS software packages we will discuss from a general usage for library applications point of view are:

- IMAGE/QUERY 3000 (Hewlett-Packard 3000 Series minicomputers),
- TOTAL (Digital Equipment Corporation. PDP-11/34 and larger minicomputers),
- DBMS-11 (Digital Equipment Corporation. PDP-11/45 and PDP-11/70), and
- MUMPS-11 (Digital Equipment Corporation. PDP-11/34 and larger minicomputers).

IMAGE/QUERY 3000

In 1973 Hewlett-Packard developed IMAGE as the first DBMS to be implemented on a minicomputer. This initial version supported only a single terminal and operated under their Disc Operating System on the HP2100 and 21MX minicomputers. IMAGE 3000 is an improved version designed specifically for the HP3000, HP3000CX, and HP3000 Series II minicomputers. The original version of IMAGE now has been replaced by IMAGE 1000 which is a compatible subset for use on HP1000 Series, HP21MX, and HP2100 minicomputers.

QUERY 3000 is a subsystem of IMAGE 3000 designed to allow non-programmers to retrieve and report data interactively from an IMAGE 3000 data base through English-like commands. These commands are translated into calls to DBMS subroutines within IMAGE 3000. We will consider both of these products here as they are companion choices of possible choice for bibliographic systems use.

IMAGE operates in both online and batch modes and is composed of four parts:

[1] A data base definition subsystem (DBDS),
[2] A data base management system (DBMS),
[3] A data base utility subsystem (DBUS), and
[4] A data base inquiry subsystem (QUERY).

Before discussing the subsystems above and discussing the full scale version of IMAGE, the significant differences between the IMAGE 1000 version and IMAGE 3000 should be highlighted. These are the number of data sets per data base, the number of detail data sets per master data set, the number of entries within a chained data relationship, data set size, and physical volume size supported, although a total data base size is only restricted by the total available disk storage.

In operation, the programmer uses DBDS to define the data base and produce a file which contains the internal system description of the data base which is called the schema. The language processor which produces this schema is the schema processor. IMAGE schema consists of data item identification as to length, symbolic name, and data type (ASCII character, integer, real); identification of groupings of data items into data sets and the relationship between these data sets, with master data sets serving as indexes to the data base and detail data sets containing the actual data; read and write privacy of each data item; and a decision as to the degree of privacy and privacy passwords.

The form of data structure resulting in IMAGE is a network type with access to detailed data sets via the master data set through linkage paths. Thus, related data items can be retrieved directly as any record can be related to any other record. Data items can be shown to be related into a network structure of master and detail data sets which are inter-related and these data items within a detail data set would have similar relationships. Figure 2., below, shows a typical network structure of the IMAGE type, but which is also employed by the next software product discussed -- TOTAL.

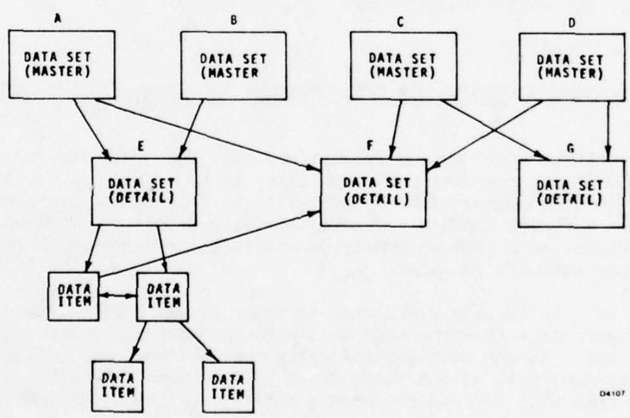


Figure 2. Network Data Structure

In comparison to DBMS-11, however, a new data set cannot be defined without impacting on existing sets. The data base build utility program must be used with IMAGE to reload the data base. Also the DBDS subsystem must be used to redefine the internal system description or schema so that the data base load utility program can load the new data set correctly, with the proper relationships expressed.

IMAGE's data base utility subsystem DBUS is composed of five programs which provide updating, restoration, backup and redefinition of the data base. Data protection and security is according to detail data set rather than individual data item. Privacy requirements cannot differ between programs. This is due to the fact that IMAGE does not employ a subschema to determine what part of the data base will be accessible by a given program.

In IMAGE a CALL verb is used to access the data base. This CALL verb obtains a library procedure called DBGET which is a GET subroutine command. A determination is thus made through examination of a parameter passed from the calling program as to which of the four available access methods is to be used to obtain the data items and sets required by the calling program. Serial access, forward and backward serial

access, chained access, directed access, and calculated access are the supported techniques.

When operating under serial access, IMAGE starts at the most recently accessed storage location for the data set called the current record. It looks at all adjacent records sequentially until the desired entry is found. In forward serial access operation, the object is to find the next higher numbered entry after the current record. In backward serial access operation, the object is to find the next lower numbered entry after the current record. Under chained access, entries have a common search key or item value and are linked together through pointers to form a chain. Here, access is then merely the retrieval of the next item in the chain currently being operated upon. For directed access, the calling program specifies the record address of the data entry where the requested data items should be located. Under calculated access, master entries are retrieved by calculating an address based on a key of some specified value.

In chained access the pointer scheme is of importance to those considering use of IMAGE for bibliographic applications. Pointers link one data set item to another. The pointers are normally paired, with one pointer referring to the previous entry in a chain and the other referring to the next entry in the chain. The last member of a chain contains a zero value forward pointer. To add a new member to the chain requires only changing the forward pointer value. Up to 16 different pointer pairs can be maintained for each data item. This permits each data item to be a member of 16 different chains or access paths. This is sufficient for most library applications, such as acquisitions, in-process file, cataloging, circulation, serials management and basic online inquiry of bibliographic files via specified indexes.

Many retrieval systems in operation today use inverted files for their storage structure. IMAGE does not support inverted file organization. However, use of the chained access and calculated address techniques is another effective manner to achieve a high degree of redundancy elimination in data coupled with excellent retrieval capabilities in files of data characterized by complex data relationships. Thus, for a system with from 10-16 terminals or perhaps a bit larger, one can achieve excellent retrieval results using techniques which do not involve inverted files.

IMAGE 3000 requires a HP3000 Series II Model 5 computer system operating under MPE II or a HP3000 or HP3000CX Model 50 operating under MPE C. A minimum of 48K words (96K bytes) of main memory and a 14.7 million byte cartridge disk subsystem are the minimal disk storage required. Run time table requirements and/or disk file control blocks will, in most systems, require some additional core but this size will be determined by the size and complexity of the particular data base.

As to system security, IMAGE provides it at the data base, data set and data item levels. A user must have access to the account containing the data base and the group in which the files containing the data base are cataloged. The programmer or data base administrator provides these levels. Users are assigned data item and data set entry privileges for reading and for writing using a class scheme having up to 63 levels. This scheme is such that a user with a level 10 access cannot access at level 9 or below. The data base administrator supplies a password to each user at each level. When the user opens the data base, he or she must supply the password as assigned as the system verifies its association with the correct level before granting access. During concurrent access by multiple users, a record lockout technique is employed to prevent two users from simultaneously modifying the same record, causing a condition called "deadlock".

The QUERY 3000 subsystem enables a user to perform interactive or batch data base interrogation. Boolean logic selection is used through the search process. Complex or frequently used commands can be stored in a command file to be reused later. Data base updating through addition, deletion, and modification is supported. The individual user may display the data base structure and perform multiple level sorts for grouped items. QUERY is also a flexible report generator, capable of handling most library output needs, except for very complex output such as a catalog card. To handle this kind of product would require the writing of a special program tailored for that purpose, but table driven to maintain flexibility for possible future change. QUERY also converts one and two word integer numbers, two word real numbers, extended precision real numbers, one word logical values as absolute numbers, ASCII character strings with or without lower case alphabetics, and zoned and packed decimal numbers. Error checking is performed prior to all conversions.

IMAGE and QUERY for the HP3000 minicomputer is a very highly rated software product by its users. It is hosted on equally reliable hardware. Designers of bibliographic systems, particularly those having complex data set and data item relationships, such as found in the Library of Congress MARC II machine-readable cataloging records coupled with diverse subsystem uses of the data base should consider this software and an associated hardware configuration appropriate to their needs.

TOTAL

This DBMS product, developed by Cincom Systems, Inc., was initially delivered to large mainframe computer users in 1969. Over 1000 installations have used the various versions of this product as rewritten for a number of different mini- and midicomputer systems. This makes TOTAL the most widely used product of its type -- partly due to the fact that versions of it are available for more computer systems than any other DBMS implemented on any minicomputer system.

TOTAL is implemented very closely along the lines of the Conference on Data System Languages (CODASYL) Data Base Task Group Report (DBTG) in its data manipulation aspects, except that rather than only supporting COBOL as the host programming language for applicatons as called for by CODASYL, TOTAL supports various application languages depending upon the particular version. These are languages which support a CALL statement such as PL-1, FORTRAN, COBOL. RPG II, and various assembly languages for specific computers.

Before proceeding further in our discussion of TOTAL a brief word about CODASYL and its DBTG

is in order. CODASYL was the informal organization which first defined the COBOL language. Their DBTG has produced a set of specifications based on ten years of research. These specifications cover data manipulation, data specification, and data structures. They are the only existing set of specifications in the field. Thus, many DBMS vendors have used these recommendations to some degree in their products as it is the currently emerging standard. The DBTG is also continuing its work to improve the facilities provided in this specification.

To use Digital Equipment Corporation minicomputers as our example, TOTAL is available to operate on PDP-11/34, 11/40, 11/45, 11/55 and 11/70 processors using RSX-11D or IAS Operating Systems in both single and multi-task versions and under RSX-11M in the single task version only. Macro-11 Assembler language, FORTRAN and COBOL are supported as application programming languages.

In this paper we will largely address the version of TOTAL for the above machines. However, TOTAL is also available for the IBM System/3, Harris Series 100 and 200, NCR Century Series, and Varian V 72, V 73, V 75, and V 76 minicomputers.

Unlike IMAGE, discussed earlier, TOTAL does not have data base security provisions built into it. If the user requires such protection, a generalized security module would have to be written which would be invoked by a user application program prior to any further action after a CALL statement to TOTAL for manipulation of the data base. Also, unlike IMAGE, no system accounting facilities are available. For bibliographic applications where a minicomputer would be dedicated to serving the users of the library's data base this is not an important feature since central processor unit time and central system features need not be accounted in the manner of large systems.

Referring again to Figure 2., TOTAL also uses a network type data structure. However, it does not incorporate any schema data description language. Its basic system operates in three phases. One phase generates the program for controlling the data base structure using a sub-schema data description language. The sub-schema defines what part of the data base will be accessible by a given program. Like IMAGE, TOTAL uses fixed length records which eases system overhead as far as manipulation but does require additional mass storage for files having greatly varying data item lengths and data set definitions. The second phase pre-formats the physical disk areas. The third phase controls access to the data base.

Two types of records -- a single entry or master record and a variable entry record are the building blocks to form a data set or collection of records. Unlike IMAGE, there is no restriction on the number of data item relationships which may be expressed or their parental relationships to data sets. TOTAL provides a bidirectional hierarchical structure technique which allows logical development of natural parent, child, sub-child relationships. The user employs the single entry master data set in conjunction with the variable data set to make these definitions.

A Data Base Definition Language is used to define the user's logical view of the data items and their relationships. This description includes the name of the Input/Output buffer, the names of all linkage fields and the structure of the records. Data item structure is very straightforward with merely name and size in bytes.

For bibliographic systems, TOTAL's main drawback is that it defines a fixed data base structure rather than a dynamic one. Records can be added to and deleted from any of the existing data sets but if new relationships are to be defined, new data sets cannot be added and disk storage expanded without at least a partial regeneration of the data base. This entails reloading all affected files. In bibliographic systems and in a minicomputer environment this could present a major problem if the system designer does not initially foresee the complexity of the relationships necessary and the time required for such reloading. Thus, for an integrated application system such as a library system serving acquisitions, in-process control, cataloging, accounting, binding, circulation and other tasks the basic subsystem specifications from a data definition view should be identified -- the data items needed by each task and their interrelationships -- even if implementation of each task module will not be immediate. This will minimize the problem of extensive, multiple reloading of the data base and its sub-schema.

Other weaknesses of TOTAL are in the area of utilities, although IMAGE is also weak in this area when compared to DBMS-11 which we will discuss next. TOTAL does not have a subsystem for inquiry and report generation such as QUERY in the Hewlett-Packard system. It will be necessary for the user to use the general system utilities supplied by the minicomputer manufacturer for backup dumping of the data base, or for restart and recovery as well as any other functions desired. If such utilities are not available, then the user will have to create these prior to any full implementation of a system.

A significant advantage of TOTAL is its highly efficient use of disk storage space and the fact that many versions of it are available on less expensive hardware configurations. It has sophisticated data structures. It uses both Basic Direct Access Method (BDAM) and Direct Access Method (DAM) within the host operating system's disk access module. Because of these points its consideration for systems on medium scale minicomputer system configurations must be seriously considered.

DBMS-11

This DBMS product conforms more closely to the CODASYL specifications than does TOTAL. The evolution of this system has been long and somewhat complex. DBMS-11 is a version of Integrated Data Base Management System (IDMS) which Cullinane Corporation originally developed from General Electric's Integrated Data Store (IDS). Thus, its functionality is one determined through many field versions over a period of almost two decades in a conventional computer environment.

Two hardware manufacturers currently offer this software product for mini or small computer use. Univac names their version DMS/90 which operates on a Univac 90/30 processor operating under OS/3

operating system. Digital Equipment Corporation's version DBMS-11, which requires either a PDP-11/45 or PDP-11/70 minicomputer system with the IAS operating system will be addressed here to parallel our other comparisons.

Due to the features of DBMS-11, currently only versions using the largest of DEC's minicomputers under their most advanced operating system are available. Thus, this is one of the most flexible, powerful, and generalized DBMS software products available in a mini- or midicomputer environment. For very large data bases with large numbers of interactive users, it may be the only commercial choice if data relationships are complex and cannot respond to any simplification to permit use of a less generalized, lower overhead system with a PDP-11/45 or 11/70 such as MUMPS-11, discussed next, or lesser hardware versions of TOTAL.

DBMS-11 employs a data description language schema and sub-schema. Both are separate from application programs. By this methodology, adhering closely to the CODASYL specifications, data base security can be provided via prohibited access and different privacy requirements for separate programs can also occur. Thus, DBMS-11 provides a greater degree of data base access security to the data item level than does either TOTAL or IMAGE. The user's password and use of the sub-schema make possible this control.

The data structure supported by DBMS-11 is again of the network type with also simple hierarchical organization within the data sets whenever it is desirable. Unlike IMAGE and TOTAL, a new data set can be defined without affecting existing, already defined data sets. Moreover, all records in a data set need not contain a common data item with a common value. This feature, coupled with the fact that variable length physical records are supported, creates a very good mass storage utilization. Of course, to manage this, there is an increase in the core storage required, the power of the input/output capability of the computer, and more elegant central processor features in addition to an operating system having many features common to an operating system found on a conventional computer system.

This DBMS product uses run-time parameters to establish, control and maintain operation. The major module at run-time is the Data Base Control System (DBCS). It acts as an executive or monitor and handles all requests from an application program prior to passing any requests to the host computer's operating system. As application programs do no physical input/output processing and reference data logically only by data item names, the DBCS uses the sub-schema module feature to translate this logical reference to a physical reference for the DBCS retrieval. Then this physical reference is passed to the calling application program.

With the sub-schema feature no data definitions exist within application programs. Each set of application programs which use the same set of data invoke the same sub-schema. Sub-schemas may overlap one another as the sub-schema for a shortened bibliographic record would represent a portion of a larger sub-schema for a full bibliographic record.

To do the translation of these logical data item name requests to a physical address request for the operating system access method supported, which is BDAM, the DBCS uses an object module called the Device/Media Control Language (DMCL). This is a separate module created by the data base administrator or system designer during data base generation. It contains the physical control blocks needed by the DBCS at run-time.

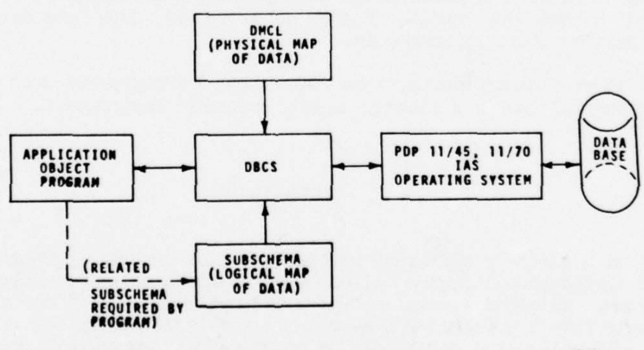


Figure 3. DBMS-11 Run-Time Processing Modules and Their Relationship to the Operating System and Data Base

Figure 3. shows the relationship between these DBMS-11 run-time modules, the host minicomputer operating system and the data base. To define the data base, the system designer talks to DBMS-11 through a Data Description Language (DDL) corresponding to the data definition language shown in Figure 1. Application programmers use the Data Manipulation Language (DML) to perform any manipulations on data once the data base is set up and loaded. These are direct implementations of the CODASYL DDL and DML specified languages.

The basic logical data entities are the data item which is the smallest logical unit of data in the data base language. In a bibliographic system, author's name would be a data item as would the dates of his birth and death. A record is the next entity which is a collection of data items. For example, a field such as Personal Name, Main Entry (Tag 100 in the Library of Congress MARC bibliographic record) could

be defined as a record composed of data items for each of the applicable subfields within this Tag. A data set is the logical relationship between record sets. To carry our example further -- all of the fixed and variable fields and subfields of MARC II bibliographic records could be shown as a data set comprising a full MARC II record. A logical collection of these records such as a given library's catalog for one of its specific collections could be defined as an AREA within DBMS-11. Data item, record, data set and Area are the four data entities common to the DDL and DML.

In addition, the DDL also defines entities that are for the use of the person defining the data base exclusively. These are not available to the application programmer who can only use the DML. These are:

- FILE which exhibits the physical characteristics of addressable mass storage. A FILE can be equal in extent to an area, a portion of an area or contain several areas. As FILE is purely a **physical** characteristic, area is purely a logical one.
- DATA BASE which consists of all of the record occurrences and set occurrences defined in a schema,
- SCHEMA which is the complete logical description of a data base using the concepts of files, areas, records, data items and data sets, and
- SUBSCHEMA which is the complete logical description of a subset of a data base which is to be defined as known to one or more specific application programs.

The application programmer may use COBOL, FORTRAN, RPG, and Macro-11 Assembler versions on the PDP-11/45 and 11/70 for writing application programs. Except for COBOL, these programs communicate with the data base by using CALL statements. For ANSI COBOL the Data Manipulation Language is used and available only to this higher level language.

Digital Equipment Corporation is pursuing the implementation of a version of DBMS-11 in an RSTS operating system configuration. No availability of this product has been announced at this writing but this development would probably bring a significantly powerful subset of features into smaller PDP-11 configurations, thus giving TOTAL a very powerful competitor. DBMS-11 is fully supported by DEC rather than its original vendor and if other versions are released this author would expect a similar support arrangement.

DBMS-11 also incorporates a generalized communications interface to permit fully interactive operation. But it does not offer a retrieval facility such as QUERY 3000. Cullinane Corporation has such a subsystem called CULPRIT but DEC does not plan to use this or offer it. Instead, DEC is in the process of developing its own query/update language.

DBMS-11 is a good choice for very large minicomputer systems, rapidly growing systems, and those data bases where a maximum of capabilities must be provided to the individual responsible for the administration of the data base. For example, DBMS-11 also employs a data dictionary module which is a reporting capability listing all logical to physical mapping, the physical distribution of files, which programs use which sub-schemas, and full schema and sub-schema listing. Neither IMAGE or TOTAL have these features.

Additionally, the most powerful set of utilities are available with the exception of the QUERY sub-system to be found on a minicomputer hosted DBMS. These include data base backup dump, data base restore, data base rollback and rollforward, examination and correction of a data base page by a data administrator, pre-sorting of records to speed up loading of a large data base, data base activity statistics, and program abort/data base restoration which is automatic.

For very large minicomputer systems, DBMS-11 and IMAGE/QUERY 3000 make these two leading systems to be examined for potential use in a library or bibliographic environment.

MUMPS-11

Although of a slightly different nature, the last program product to be examined here is a combination DBMS and application language called MUMPS-11 which means Massachusetts General Hospital Utility Multiprogramming System. Originally developed on a DEC System 10 medium scale computer, versions are now available for the whole PDP-11 Series. For example, a PDP-11/10 system can support from four to six users. The PDP-11/34 based MUMPS-11 system can handle up to 32 users. Moreover, versions of MUMPS are now available for Data General Nova and soon will be for several other minicomputers. The American National Standards Institute (ANSI) has voted in September 1977 to adopt a standard MUMPS, making this the third computer language to be so standardized.

MUMPS-11 is both a data base management system using hierarchical data relationships and an application language. A user of MUMPS must construct all of the programs needed for an application using this language along with the DBMS facilities of the system. MUMPS is optimized toward ease and speed of access to data and application code development.

In the United States, the National Library of Medicine has implemented many of its internal management functions such as cataloging using MUMPS. They have experienced very short development times to bring rather complex applications and files to a working state. Washington University School of Medicine Library in St. Louis, Missouri has installed their updated version of their networked serials management system called PHILSOM III using a MUMPS PDP-11/40 configuration. There is no doubt that MUMPS easily lends itself to the problem of bibliographic systems.

For users that do not require the security, the full network data structure, and the large number of utility aids that DBMS-11 offers, MUMPS-11 should be considered. The main drawback to MUMPS for most minicomputer application systems is the requirement to use the MUMPS language for all applications,

although for the smaller systems particularly, its facilities are certainly adequate for the development of workable systems for libraries. Moreover, MUMPS language enhancements stem from the fact that two basic dialect versions have appeared. The A-based dialect more closely resembles standard MUMPS. The B-based dialect, developed and maintained by Medical Information Technology, Inc. and called Miis has features and options not found in the standard MUMPS. Here we will look at the standard MUMPS as implemented on the PDP-11 Series minicomputers.

As previously mentioned, the file structure of MUMPS-11 is hierarchical, with any number of levels permitted. Up to two million nodes may exist at any level. Each node in a file may contain a pointer to a lower level or contain data or contain both. The higher levels usually consist of the most significant digits of record identification numbers of a fixed length determined by the file size. The level consisting of data or data and a pointer to a yet lower level does not contain the identification number as it was stored in the highest level.

With this type of structure, MUMPS-11 reads the first node, finds the pointer to the second, reads the pointer to the third node or the data and so on down to the last level node encountered. A disk access is required for each level, but in most systems three to four levels are commonly used, which means that access is still very fast -- typically on the order of 100-120 milliseconds for bringing the desired record into core by a calling program.

Another feature of support to the file capability of MUMPS is a sequential disk processor which provides a facility to allocate, at system generation time, a portion of available mass storage to this processor. MUMPS-11 programs view this area as a contiguous area of byte addressable storage blocks without any structure. These blocks enable compatible PDP-11 DOS files to be built under timesharing for later batch processing under the DOS operating system. Another use of this feature is as a spooling area for output to a line printer. This area would be called by a printing program residing in a small user partition of main memory. This is more efficient than having each application program request the line printer individually.

Since MUMPS-11 is a rather unique system of both a simple DBMS and a language, a brief description of the basic MUMPS language is in order. MUMPS language commands are relatively few but very powerful. Twenty-five commands are organized into four main classes. Figure 4. below shows these commands to enable their comparison to other languages with which the reader may be familiar. Figure 5. illustrates the MUMPS operators.

GROUP 1. ASSIGNMENT		EXAMPLES
SET	Set a variable equal to an expression	SET A=B+5*C
KILL	Deletes local variables	KILL A
XKILL	Deletes all but the specified variable	XKILL B
GROUP 2. CONTROL		
GOTO	Unconditional transfer	GOTO 5.4[illegible]
DO	Subroutine call	DO 5
IF	Conditional branch	IF A>B DO 5
FOR	Loop Control	FOR I=1:1:10 DO 7
CALL	Subroutine call to program	CALL PROG.
OVERLAY	Overlay program. Does not provide return.	OVERLAY PRZ.
START	Starts the execution of a related program in an available partition	START JOB
QUIT	Prematurely terminates a loop	Q
HANG	Timed delay	H 5
HALT	End of program	HALT
GROUP 3. I/O		
TYPE	Output to principle device	T "Author", AUT
READ	Input from user	R "Enter Author", AUT
PRINT	To output special control character	P REW
WRITE	Command to write out program	W ULS
ASSIGN	Establish ownership of a peripheral device	A LP
UNASSIGN	Return a device	U MT
GROUP 4. SYSTEM CONTROL		
MODIFY	Correct a program step	M 1.5:/OLD/NEW/
ERASE	Delete a program step	E 1.5
LOAD	Load a program	L PG1
FILE	File a program	F PGZ
BREAK	Temporary halt for debugging	B 1
VIEW	Examine or change core location	V BU1

Figure 4. MUMPS-11 Commands.

SYMBOL	MEANING	CLASS
+	Addition	NUMERIC
-	Subtraction	
*	Multiplication	
/	Division	
<	Less than	RELATIONAL
>	Greater than	
=	Equality	
&	And	LOGICAL
!	Or	
'	Not	
[	Contains	STRING
]	Follows	
?	Pattern verification	

Figure 5. MUMPS-11 Operators.

Note that powerful string processing or pattern matching operators may be used with the MUMPS-11 commands. For bibliographic systems, this is one of the strengths of the MUMPS approach to system design. Additional functions to be used in the construction of MUMPS programs are shown in Figure 6. Again note the string functions available.

MUMPS-11, as a system development tool has the advantage of simplicity in comparison to the other DBMS products discussed here. The application development task is quite rapid in the program coding, debugging, and testing stage when compared to their development without any DBMS philosophy at the heart of the application design.

Of interest to libraries considering MUMPS-11 is the prospect of locating some already developed programs through the MUMPS Users Group (MUG). For example, the U.S. Department of Justice, Drug Enforcement Administration in Washington, D.C. has developed a system which is in operation under its initial version.[1] This system is PATHFINDER I which resides in a PDP-11/45 operating under MUMPS-11. It is programmed in MUMPS and uses the file organization facilities of MUMPS but provides a full DBMS capability with its own Data Definition Language (DDL) and Uniform Data Language (UDL). PATHFINDER I is both a powerful DBMS and an

FUNCTIONS	SYMBOL	FUNCTION	EXPLANATION
NUMERIC	$C	CREATE	Creates a unique numeric value from 3 characters
	$D	DEFINE	Checks data type of a variable
	$F	FIND	Finds the character position in a string
	$H	HIGH	Obtains the next higher element in an array
	$I	INTEGER	Truncates decimal fractions to integers
	$L	LENGTH	Calculates the length of a string
	$M		Floating point function
	$N	NEXT	Obtains next step number
	$Q		Next physical element in an array
	$R	ROOT	Square root
	$V	VIEW	Returns the contents of a core location
STRING	$A	ALTER CASE	Converts upper case to lower case and vice versa
	$E	EXTRACT	Extracts character from specified location in a string
	$P	PIECE	Extracts fields from a string
	$S	STEP	Obtains contents of a step
	$T	TEXT	Converts numbers to text

Figure 6. MUMPS-11 Functions.

excellently thought out inquiry or retrieval system. Although this system is in the public domain, no availability of it to other users has been announced. In fact, PATHFINDER II, to be greatly enhanced, supporting multi-file searching, interfaces to other computer systems, interactive graphics/plotting mode for data analysis, is now under development. Its implementation is planned for late 1978 with the possibility of a version able to operate under another DEC operating system, such as UNIX developed by Bell Telephone Laboratories for the PDP-11/45 or 11/70. However, at this writing a firm decision on which advanced operating system would be employed had not yet been made.

CONCLUSION

The advantages of the DBMS approach to bibliographic systems design, in the opinion of this author, outweigh the traditional single application development approach, for all but the very simpliest and smallest systems. The multiple use of data items, the rapidly changing user requirements, the prospect of greater application of the computer within the lower cost hardware environment of the minicomputer, all point toward the desirability of the DBMS approach. However, with this approach goes a concomitant responsibility toward the function of data base administration. This function must reside in an individual thoroughly cognizant of the user's application environment and the facilities of the chosen DBMS to handle the mapping of logical user's views of data to the physical view dealt with through the DBMS. If this responsibility, from the outset of the project, is not recognized, the DBMS installation will not be carried out successfully.

This paper merely attempts to familiarize the reader with some of the general aspects of these systems, offer some comparative data, and some comment related to the use of these products within a bibliographic or library management system within a minicomputer environment. With the caveat about the importance of the data administrator function, each of the systems considered here could be a logical choice for use in a library system given the careful analysis of that library's needs, the hardware able to be supported in the library's geographic location, the funds available, and the general system requirements upon which compromise cannot be tolerated.

REFERENCE

1. Levine, Emil H.; Steve E. Rodgers; Stanton E. Mintz. "PATHFINDER I - A Minicomputer DBMS - Low Cost High-Powered Computing." A Poster Session at the 40th Annual Meeting, Chicago, IL of the American Society for Information Science, September 26-October 1, 1977. Abstracted in Proceedings of the ASIS Annual Meeting, Volume 14. New York, Knowledge Industry Publications Inc. for ASIS, 1977. ISSN 0044-7870 ISBN 0-914236-12-1. p. 46-47.

FUTURE PROSPECTS FOR MINICOMPUTERS

Frank V. Wagner
Senior Vice President
Informatics Inc.
21031 Ventura Boulevard
Woodland Hills, California 91364

SUMMARY

The minicomputer revolution has reached the point of no return. Decentralized computing will be a very important factor in the thinking of managers of Information Systems. The point of view of a businessman is presented, which strips the mystique from computers and puts them in perspective as an operational tool.

As an example, a very simplified and inexpensive minicomputer system is described. It assists with catalog maintenance and is within the budget of most medium-sized libraries. The trends which have brought this about are presented and analyzed. The problems caused by the dead hand of Grosch's Law (and its lesser known corollary) are presented. A counter-principle, "The Principle of Decentralized Computing" is proposed to replace it.

INTRODUCTION

In contrast with most of the speakers at this meeting, my profession cannot be described by the distinguished title of "information scientist" or by the older, honorable title of "librarian." My profession is Management. Originally an engineer, I soon became a manager of engineering activities, then a manager of data processing activities, then finally manager of the activities of a corporation in the information processing industry. Consequently, I address you from the point of view of a businessman — one who is concerned with the most efficient way of increasing the productivity of an enterprise by increasing efficiency and reducing costs. I am here to forecast the future, so perhaps the title of my talk should have been, "Prophecy and Profitability."

The Art of Prophecy

In order that a prophet avoid a credibility gap, he must first establish his bona fides by indicating some past success in the prophet business. I have been a manager in the computing business for a very long time, since the late 40's. Part of that experience as a manager involved making many forecasts, most of which I had to back up with my employer's money.

In order to lend an air of precision to my prophecy, I must first define what I mean by saying that something happens. Data processing is a continuous spectrum of events blending into one another. But it is possible, by standing back a little to identify definite milestones.

There are two important types of milestones. The first of these is the "Point of No Return." It is the time at which 50% of the leading installations are commited to a new modus operandi, and the trend is up. To some extent this concept is intuitive — for example, the concept of "leading installations." These are generally the large data processing organizations, the ones whose staff members are prominent in the professional world, the ones who pioneer new things and who are always working in the forefront of the field. When 50% of these installations are committed to some new methodology, then the art has reached the point of no return. Thereafter, anyone who is not seriously considering that concept is in danger of falling behind and had better start to worry.

The second milestone is the "Fait Accompli." At that point the modus operandi is no longer new or controversial. Not to be committed to it now requires justification.

Previous Prophecies — and How They Flew

With those two steps in mind, consider the track records that I display in Table 1. These predictions have been recorded in the archives of my employers and of SHARE, the famous IBM users group.

Table 1. Early Prophecies

Year Prophecy Made	Subject	Point of No Return Predicted In	Point of No Return Actual	Fait Accompli Predicted In	Fait Accompli Actual
1954	Higher Order Languages	1958	1960	1960	1962
1954	Time-Shared Data Capture	1957	Never(?)	1960	Never(?)
1955	Operating Systems	1957	1957	1960	1959
1955	Numerical Control of Machine Tools	1957	1958	1958	1959
1957	Universal Computer-Oriented Language	1960	Never	1966	Never
1960	Spooling	1962	1963	1966	1964
1962	Virtual Memories	1964	1967	1970	1972
1962	Wide-Spread Graphic Output	1966	1966	1970	1972

Out of my eight early prophecies, six came to pass. However, in most cases they lagged my predictions by one or two years.

In 1965 at a symposium sponsored by the University of California at Los Angeles and Informatics Inc., I made a number of similar prophecies about the transition from batch processing to transaction-oriented on-line processing. These were recorded in the book published as proceedings of the conference (Ref. 1). Some of them are shown in Table 2.

Table 2. 1965 Prophecies

Subject	Point of No Return Predicted In	Point of No Return Actual	Fait Accompli Predicted In	Fait Accompli Actual
Remote Job Entry Stations	1968	1968	1970	1970
Time-Shared Program Development	1970	1967	1975	1972
Store & Forward Data Transmission	1971	1970	1974	1973
Basic Processes of the Enterprise	1972	1975	1978	?

Out of these four prophecies, three came to pass earlier than I predicted.

Current Prophecy

What has all of the foregoing to do with "Future Prospects for Minicomputers" as it applies to you and your work? The same methodology is applicable that I used to make those predictions, namely: (1) Extrapolate the technology, (2) analyze fundamental needs of the users of data processing, and (3) MOST IMPORTANT, observe the way people think and act in their use of computers, and predict when they will be able to apply to their real needs the technology that will be available to them.

The application of such a methodology leads me to quantize for the first time a prediction that I have been making qualitatively for the last several years:

"Decentralization of data processing facilities is inevitable."

Within a relatively short period of time, certainly less than ten years, a large majority of the functional units in any enterprise which have a need for data processing will have their own computer to do it, dedicated solely to a single function which is the responsibility of that unit. Such decentralization of both the physical facilities and the responsibility for them is inevitable. Using the terminology that I explained above, the "Point of No Return" (when 50% of the leading installations have adopted it) will occur by 1980. The "Fait Accompli" milestone (when you have to justify not operating that way) will occur by 1985.

DECENTRALIZATION IS INEVITABLE

I will now elaborate on the thinking that leads me to such a conclusion. The driving force is not the technology — it is the behavior of computer users. Data processing people have glorified their toys so successfully that they have clothed them in a mystique which obscures their only purpose — to serve people. To illustrate the problem, let me tell you a couple of stories that I have used many times.

A while ago, as I was skimming through a financial publication, I was approached by a friend with a philosophical mind. I commented on the interesting item I had been reading about the remarkable growth of the consumer power tool industry. It seemed that astonishing records were being set for the sale of one-quarter inch electric drills. I expressed my surprise that there were so many people who wanted one-quarter inch drills. My friend observed, "I don't believe there is any real demand for them." When I asked how he could come to such a conclusion in the face of the evidence, he pointed out: "People do not want one-quarter inch drills. What they really want are a great many one-quarter inch holes."

Computer professionals, in their saner moments, will acknowledge that computers are only tools to enable us all to achieve our real objectives. But unfortunately, computer professionals usually talk and act as though the computer is an end in itself. However, the realization is slowly coming into focus that we don't really want one-quarter inch drills; we just want a lot of one-quarter inch holes. Fortunately, technological improvements which are driving down the cost of computing will make it possible for users to approach the subject a little more rationally than we have in the last twenty years.

Summary of Cost Performance Trends

Fig. 1 has been derived from many sources, including Withington (Ref. 2) and Phister (Ref. 3). It shows that dramatic reductions began in 1960 in cost performance trends for the elements of computing systems. Major cost reductions in central processors have been accomplished by 1975, and there will be continuing reductions through 1985. Reductions in communication costs, on the other hand, lag far behind all other types of hardware costs. The cost of software development has been progressing nicely, but dramatic improvements can only be achieved by using software products. The past and future improvements are:

Table 3. Reduction in Costs

	1975 Costs As a Fraction of 1960 Costs	1985 Costs As a Fraction of 1975 Costs
Hardware		
Processing and Internal Storage	0.005	0.20
Fast Access Mass Storage	0.02	0.10
Common Carrier Communication Lines	0.61	0.53
Software		
Custom Development	0.28	0.47
Software Products	0.06	0.33

The Efficiency of People

Because computers have been, in the past, so terribly expensive, we have forgotten something important. In most organizations, data processing costs tend to be somewhere from 0.5% to 2% of total costs. Salaries and other costs of the people doing the basic work of the enterprise are between 40% and 90% of its total costs. The efficiency of the total enterprise is the efficiency of the global system of people doing the necessary work, assisted by computers. Yet, at meetings of computer people, the discussions might lead one to believe that all the work was done by computers. Serious papers are published on optimizing the hardware/software sub-system, leading the unwary to believe that such optimization would solve the whole problem. To illustrate the fallacy of optimizing a sub-system, let me tell you a parable.

Just suppose that typewriters had never been invented. All the secretaries in your office are producing letters and reports by writing them out very neatly in longhand with ballpoint pens. Suddenly Olivetti makes a dramatic announcement — it has invented the typewriter! Next morning, at every large corporation, an Olivetti typewriter salesman has an appointment with the Vice President for Office Services. Let us see what happens, for example, at Philips Gloelampenfabrieken. The salesman suggests that Philips buy a $600 typewriter for each secretary. This is a revolutionary proposal. Purchasing is called in. Internal Consulting is summoned to a meeting. Operations Research is charged with investigating the concept, and conducting a feasibility study. The results show that the average use of a typewriter by a secretary would be 1.1873 hours per day, and that the productivity of secretaries would increase by 325.26%. It is strongly recommended that the corporation convert to typewriters. It is recommended that enough typewriters be procured so that each is loaded 4.7492 hours per day. Allowing for down time and assuming that most overloads can be handled by overtime, each four secretaries will share one typewriter.

The recommendation is adopted. The typewriters are delivered, and training begins. Soon all secretaries are mechanically proficient; Purchasing cancels all orders for ballpoint pens; and the system is cut over to document production by typewriter only. Productivity is very low the first week. It is worse the second week, worse the third, and by the end of the month, the backlog of letters and reports has reached alarming proportions. Most executives are spending a good deal of their time consoling tearful secretaries who complain that they cannot get their work done because they cannot get access to a typewriter. An emergency meeting is held. Only two viable alternatives present themselves — go back to ballpoint pens, or get a typewriter for each secretary! Operations Research does a fast study, and concludes that spending four times as much for the typewriters will be paid for many times over by the increase in productivity. The recommendation is accepted, the secretaries live happily ever after, and Philips increases its dividends.

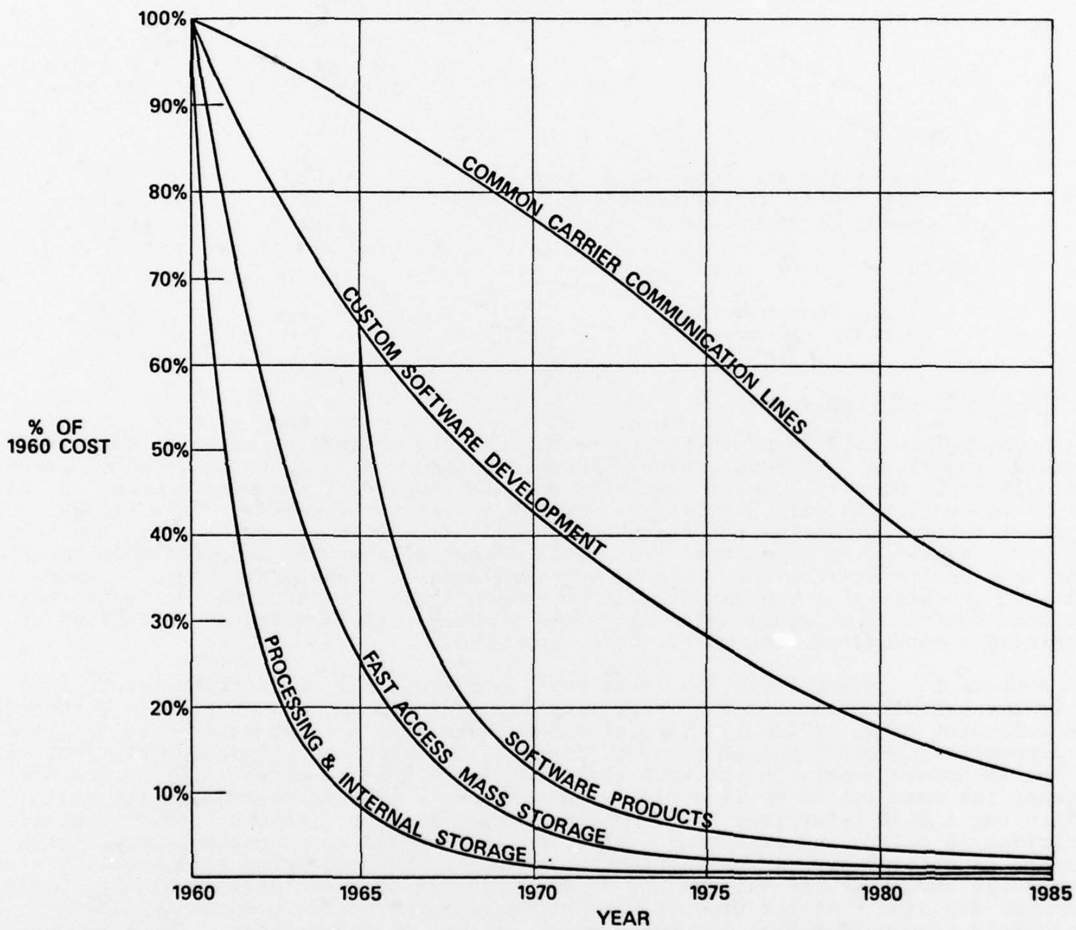


FIG. 1—RELATIVE COSTS

The moral of the parable is clear. *Efficiency of people can vary over a wide spectrum.* The efficiency of people can seriously be impaired by frustration — "Nobody lets me get my work done without interference." Conversely, productivity seems to be at its highest when the worker has full control of the tools he needs. Many of you have had experiences with central data processing similar to those of the secretaries with the shared typewriters. What would happen if you had control of your own computer? If anything went wrong, you would look around for someone to blame, and see no one but your own unit. You would roll up your sleeves and solve the problem.

Of course, the moral of my parable would not be applicable if typewriters cost $600,000 apiece, instead of $600. Until ten years ago, computers were too expensive to consider decentralizing control of them. But today their prices, including software, are approaching the low level where it is foolish to sub-optimize computing costs at the expense of the costs of the people who really do the work.

Trends in Leading Installations

A number of pioneers are proving the validity of the decentralized concept. Canning (Ref. 4) records several interesting case histories. One of them, Citibank (First National City Bank of New York), is the largest bank in the United States in terms of net income. Through the 1960's it was a leading example of centralization of computing facilities. Beginning in the early 70's it did a complete about-face. It recognized the importance of giving operating units their own computers, and began a massive program of decentralization. White, the architect of this revolution, tells the story in (Ref. 5). Another example is Hughes Aircraft, a multi-billion dollar aerospace firm, which embarked in the early 60's on the consolidation of all data processing facilities and responsibilties into a super-centralized facility. In a fascinating public confession, Reynolds (Ref. 6) recants all of the dogmas that he believed in the 1960's and espouses the cause of giving each functional unit total control of its own data processing, including the physical computer.

Distributed Processing

I have been using the term "decentralized" in order to emphasize local autonomy and control of as much as possible of the data processing function. In data processing literature, you will today see much more frequent reference to "distributed processing." Canning (Ref. 4) uses the terms interchangeably. However, most people use "distributed processing" only to describe the trend toward incorporating "intelligence," (i.e., processing capability) into the remote terminals. There is a definite growth in the use of such intelligent terminals, which incorporate minicomputers. More and more, the processing which *can be done remotely is done remotely*, and only that which requires a central computer is transmitted to and from the central site. This concept, known as "distributed processing" is rapidly gaining in popularity. It is a hybrid system, partially decentralized. It is fostered by IBM and by the management of centralized data processing as long as they can retain control over it. However, since the cost of communication is not coming down as rapidly as all other hardware, it will soon become obvious that the remote processing done by such "distributed processing" will require a data communications link to a central site only in rare instances. I expect to see such umbilical cords cut with increasing frequency in the next decade. Then there will be no reason not to divorce the operating unit from the control of central data processing management.

A Decentralized System for Use in Libraries

An example of a minicomputer system operating in a decentralized mode is the MINI-MARC system now in use in the library of the United States Department of Energy. It was developed to increase the productivity of the cataloging staff of a library which uses, as its source information, the data issued in special format for each book by the Library of Congress. However, for this particular library's own catalog, it is not appropriate to use the information in its raw form. The cataloging staff eliminates some of the data, and adds additional data, using MINI-MARC.

The system has the following fundamental characteristics; it:

- Is a stand-alone unit which requires no telephone line hook-up or connection with a main-frame computer.
- Provides access to the records on the Library of Congress (L.C.) MARC Distribution Service tapes.
- Handles all MARC formats: Monographs, serials, films, maps, manuscripts, and music (when issued).
- Provides record look-up by main entry, by title, and by L.C. card number.
- Displays a full MARC record — complete with fixed fields, variable tags, indicators and subfield codes.

- o Provides the ability to revise and/or add local information to MARC records.
- o Allows a library to build a working file of selected MARC records.
- o Allows a library to input original records into the working file.
- o Constructs working files which can be used as input for BIBPRO IV, a set of programs for a central 360/370 to produce thereon catalog card sets, book catalogs, COM, KWOC index, etc.

Fig. 2 is a simple schematic of the MINI-MARC system. It consists of the following:

<u>HARDWARE</u>

Basic: A minicomputer configuration made up of the following components: Processor (CPU), CRT Console, and Floppy Disk Storage Device.

Optional: Printer, Magnetic Tape, and Acoustical Coupler.

<u>DATA BASE</u>

Over 600,000 records from the MARC tapes written on 300 to 400 floppy disks. New floppy disks are provided continually to update and keep the data base current with the MARC subscription service.

<u>SOFTWARE</u>

Basic: A program which allows access to a record by L.C. card number; displays the full record; allows changes to be made to a record; allows local data to be added; writes the record onto a workfile on a floppy disk; and allows a complete record to be keyed in and written onto the workfile.

Optional: A program to compose a catalog card format and print it on an attached printer. A program to write records onto a 9-track magnetic tape in MARC II Communications Format.

<u>How Does It Work?</u>

A book can be searched by main entry (author), by title, or by Library of Congress card number. There are presently two indices in hard copy:

1. L.C. Card Number Index — Lists the card number and a floppy disk number.
2. Author/Title Index — Lists authors and titles interfiled into one alphabetical sequence. Each author or title entry lists the floppy disk number and the L.C. card number.

The floppy disk cited is pulled and mounted. The L.C. card number is keyed in and the system searches the floppy disk and displays the correct record in less than three seconds.

The record is displayed in ascending tag order starting with the L.C. card number, followed by the fixed field data. The forty characters of the latter are broken up into nine logical units and each is identified with a label or a "prompter." The variable fields follow, each with complete numeric tag and indicators and subfield codes inserted within the text of the field. The prompters, tags, indicators, and subfield codes are displayed in lighter face type than the text of the record, to allow greater ease in reading and reviewing the record.

The record can be manipulated through use of the keyboard and function keys. These activities include: Insert a blank line, delete a line, replace characters, insert a new field, display previous record. This allows a record to be revised or local data added. The record can be written onto the working file (user file) disk in the revised form. <u>The pure MARC record cannot be altered in the data base proper</u>.

An original record (not based on the L.C. record) can be entered. The CRT displays prompters for building a MARC-like record. The record is also written onto the working file disk.

The working file, or user's file, serves as a permanent record of the library's accessions. The user's file can be used to produce output products such as proof lists, catalog cards or book catalog pages.

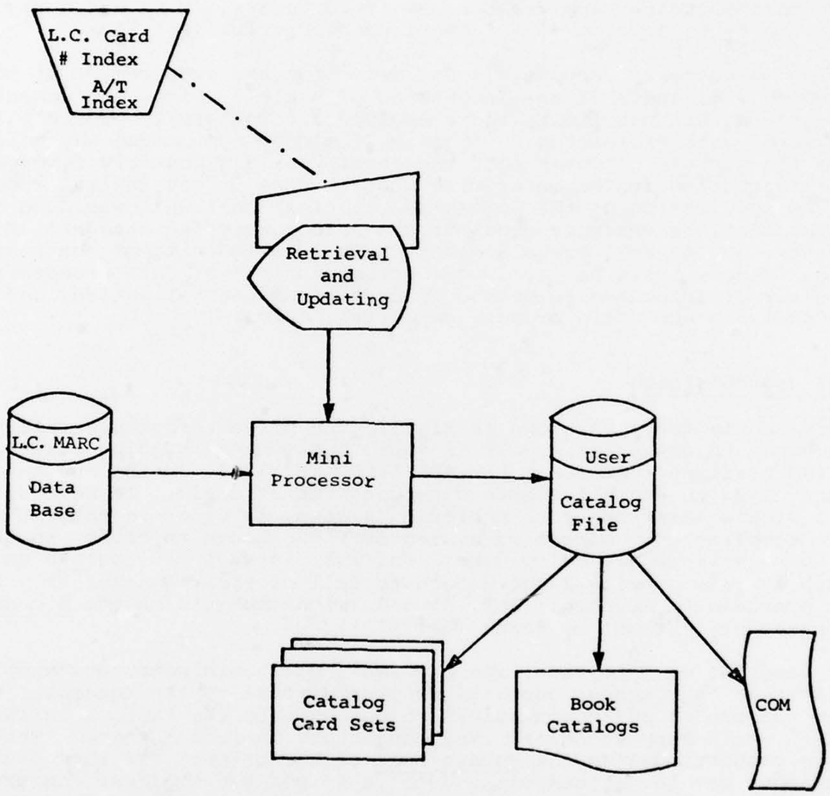


FIG. 2 - THE MINI-MARC SYSTEM

Ten years ago such an application would have been done on a central computer. Today it is nonsense to do so — but most data processing managers would not admit it because of the dead hand of Grosch's Law.

THE MYTH OF GROSCH'S LAW

In the first quarter century of the computer age, from 1950 through 1975, a number of generations of machines were developed. As each major computer was offered for sale, it could perform more work for less money than its predecessors. Dr. H.R.J. Grosch, the current president of the Association for Computing Machinery, was the first one to popularize a formal description of this phenomenon. It became known, humorously at first, as Grosch's Law, formulated as follows: "Throughput capacity of a computer is proportional to the square of its price." A number of more formal documented studies of this phenomenon have confirmed that the "Law" was indeed valid for the large computers of that era when operated in batch mode; e.g. Littrell (Ref. 7), Solomon (Ref. 8). (Cynics, however, pointed out, in Oldehoeft and Halstead (Ref. 9), that the Law was not fundamental because of the economics of engineering development, but only accidental because of IBM pricing strategy.)

In the same era, there was another phenomenon occurring. The users of such large machines were making greater and greater use of computers. More and more computer applications were being developed, as described by Patrick (Ref. 10).

Data processing management, responsible for selection and procurement of computing equipment, became well aware of the "economies of scale." Since the demand for capacity seemed to be growing without limit, there evolved a corollary to Grosch's Law, that I call the "Dogma of Data Processing." It is: "The most economical way to do computing is to acquire the largest computer that the enterprise can possibly foresee a need for; therefore, all computing in the enterprise must be done in one central computing facility." The application of the Dogma made it clear that what was good for data processing installations was very good for the data processing manager! His prestige, power, authority, and salary were also proportional to the size of the installation that he ruled. Hence, there has evolved a strong "union" of data processing managers who quote the gospel according to Grosch in defense of centralization, and treat as heretics the radicals who would propose decentralization.

The Repeal of Grosch's Law

As costs decreased in the ways shown in Fig. 1, the price-throughput relationship of Grosch's Law began to disappear. Half in jest, but with astounding foresight, Adams (Ref. 11) first headlined "GROSCH'S LAW REPEALED" in 1962. On the one hand, the very large machines began to encounter some dis-economies of scale. As more and more work was done on a single mainframe, the operating system, in order to sort it all out, got more and more complicated. Operating system overhead began to climb, so that less and less useful work was actually being done. Patrick, in (Ref. 10) points out the possibility of such systems growing large enough to fall of their own weight. Further testimony is provided by Reynolds (Ref. 6) and (with emphasis on the dis-economies of data base management systems) by Frank (Ref. 12).

On the other hand, as we have seen, the new small cheap minicomputers simply did not follow the law that "throughput capacity is proportional to the square of the price." Although I am unaware of any computations to demonstrate the fact, I believe that the data since 1965 would support Adams' 1962 conjecture (Ref. 11) that: "Throughput performance is proportional to the square root of the price." If this conjecture is true, the cheapest way to get computing done is to get the smallest computer that can perform a particular application. The $4.95 hand-held calculator may be an illustration of the latter "law."

THE PRINCIPLE OF DECENTRALIZATION

I believe that we have entered an era where guidance in acquiring computing power can be derived from a new Principle of Decentralization:

> "If any functional group, smaller than 30 people, requires computer assistance, it is better for the total enterprise that those people have exclusive use of their own computer — provided that the computer, big enough to do the job properly, will be loaded to over 10% of its capacity."

The underlined words are vital to the application of the principle.

The group must be smaller than 30 people [illegible] number that should decrease in the future [illegible]. It was chosen so that the cost of a dedicated [illegible] of the total personnel costs. (Surely, adding a computer [illegible]) Moreover, if the group is [illegible] than one kind of work to do, and thus, the [illegible] (Ref. 13) [illegible] have a small central computing [illegible] of the advantages of

decentralization. Thus, the importance of the words, "exclusive use" — which cannot happen unless the group is smaller than 30 people. If such a group performs several functions, it should consider one computer for each function.

The computer must be big enough to do the job properly. There are many problems which simply cannot be handled by today's small computers. Nuclear reactor design, numerical prediction of global weather, maintenance of reservations a year ahead for all of the airline seats in the world, instant retrieval from massive data bases, are BIG problems. Such applications are only feasible today on a computer so expensive that it is only practical if it is shared by many groups of users. I venture no predictions as to the speed with which technological improvements will catch up with these applications. Up until recently, I believed that large data bases would be the last application to become adaptable to a dedicated small computer. But a careful analysis shows that such is not always the case. For example, data bases can frequently be segmented and distributed to the place where the segment is used. "Instant" updating may be unnecessary; Canning (Ref. 4) notes that a large insurance policy file has been distributed among branch offices, each of which services a distinct small group of policy holders.

The last part of the principle of decentralization speaks to the utilization of such a dedicated computer. It stipulates that the computer should be loaded to over 10% of its capacity. Logically, of course, the loading is immaterial, as long as the productivity of the group is increased enough to pay for the computer. However, I have added this requirement, without any analytical justification for the selection of the 10% level of use, because of my belief that, if it is to be successful, the assistance of the computer to the group should not be for some trivial application, but should participate in the main-stream activity of the group.

CONCLUSIONS

It is now clear that Grosch's Law has been repealed by technological advances. IT IS NO LONGER TRUE THAT THE MOST COST-EFFECTIVE WAY TO DO DATA PROCESSING IS ON A LARGE CENTRAL COMPUTER. The new "Principle of Decentralization" is a useful guide for a small group to use in evaluating whether to use central data processing or whether to acquire its own dedicated computer.

However, certain powerful forces will delay for many years the inevitable growth of decentralization. These are the current investment of large amounts of money in central data processing installations and in the organizations built to support them, and especially the vested interests of their management and of IBM's manufacturing capability. Nevertheless, the "Point of No Return" is 1980, when the majority of leading installations will be decentralizing. By 1985 decentralization will be a "Fait Accompli" and you will have to justify to your boss why you are sharing the use of a central computer.

REFERENCES

1. Wagner, F. V., "How the Computing World Went On-Line," Gruenberger, F., Editor, Academic Press and Thompson Book Company, 1967, pp. 183-195.

2. Withington, F. G., "Beyond 1984: A Technology Forecast," Datamation, Jan. 1975, pp. 54-73.

3. Phister, M. "Data Processing Technology and Economics," Santa Monica Publishing Company, 1976.

4. Canning, R. G., "Distributed Data Systems," EDP Analyzer, June, 1976, Vol. 14, No. 6

5. White, R. B., "A Case Against Large-Scale Computers," Infosystems, August, 1975, P. 34.

6. Marienthal, L. B., "Small Computers for Small Businesses," Datamation, June, 1975, pp. 62-78.

7. Littrell, R. F., "Economies of Scale in the IBM 360 and 370," Datamation, March, 1974, pp. 83-88.

8. Solomon, M.B., "Economies of Scale and the IBM System/360," Communications of the ACM, June, 1966, pp. 435-440.

9. Oldehoeft, A.E. and Halstead, M.H., "Maximum Computing Power and Cost Factors in the Centralization Problem," Communications of the ACM, Feb. 1972, pp. 94-96.

10. Patrick, R.L., "Decentralizing Hardware and Dispersing Responsibility," Datamation May, 1976, pp. 79-84.

11. Adams, C.W., "Grosch's Law Repealed," Datamation, July 1962, pp. 38-39.

12. Frank, W.L., "The Ten Great Myths of Software," Address to Conference on Software, American Institute of Industrial Engineers, Washington D.C., December 1977.

13. Dorn, P.H., "The Trouble With Minis," Datamation, May 1976, p. 82.

SUPPLEMENTAL BIBLIOGRAPHY FOR LECTURE SERIES 92
THE USE OF INEXPENSIVE MINICOMPUTERS
IN
LIBRARY APPLICATIONS

DATA BASE MANAGEMENT ON SMALL COMPUTERS
National Aeronautics and Space Administration. Marshall Space Flight Center, Huntsville, Ala. Computer Sciences Corp., Falls Church Va.
August 24, 1974 February 6, 1978
NASA 76K11834
Algorithms, Census, Computer Networks, Computer Techniques, Computers, Data Bases, Data Compression, Data Management, Data Processing, Display Devices, Imagery, Information Dissemination, Information Management, Information Systems, Land Use, Man Machine Systems, Management Information Systems, Minicomputers, Optical Data Processing, Urban Planning, User Requirements.

A MINI-COMPUTER BASED INFORMATION SYSTEM
by McIvor R.A.
Defence Scientific Information Service, Ottawa (Ontario)
June 1974
NASA 76N71735
Information Retrieval, Management Information Systems, Minicomputers, Computer Programs, File Maintenance (Computers), Selective Dissemination of Information.

FEASIBILITY OF EXECUTIVE MIMS ON INTERDATA 80
by Bauer, M.F.; Irani, K.B.
Michigan Univ., Ann Arbor. (Dept. of Electrical Engineering)
October 1973
NASA 74N72927
Feasibility Analysis, Management Information Systems, Medical Services, Minicomputers, Algorithms, Data Processing, Data Storage, Information Management.

REPORT TO THE BRITISH LIBRARY (RESEARCH AND DEVELOPMENT) ON THE USE OF THE GRANT S1/51/44 FOR THE EVALUATION OF ON-LINE DATA CORRECTION METHODS USING A MINI-COMPUTER
Oxford Univ. (England). (Bodelian Library)
March 1977
NASA 77N32990
Two Procedures for editing bibliographic data in the Bodelian Computer System are discussed. A device driver was prepared and incorporated into the operating system so that a Visual Data Unit (VDU) can be used as a very fast teletype, with display and modification of the data performed by the program. In the second technique, the hardware editing features of the VDU acting upon data stored in the VDU memory are used.

Editing Routines (Computers), Grants, Great Britain, Libraries, Minicomputers, On-Line Programming, Computer Graphics, Cost Analysis, Data Processing Terminals, Punched Tapes, Teleprinters.

MINI-COMPUTERS AND BIBLIOGRAPHIC INFORMATION RETRIEVAL
by Hyman, M.; Wallis, E.
British Library Lending Div., Boston SPA (England)
July 1976
NASA 77N22001
The characteristic features of minicomputers and their differences from the larger machines on which most bibliographic information retrieval systems have so far been based are discussed. The design and operation of mini-computer-based information retrieval systems are described, and the advantages and limitations of alternative approaches are explored. Topics include the relative merits of on-line and batch processing for the various components of the system, and the selection of storage media, file structures, terminal equipment, enquiry languages, and programming languages. The technical features and costs of four model systems illustrate the potential use of minicomputers. A glossary lists the computer and information retrieval terms used in the report.

Bibliographies, Information Retrieval, Libraries, Minicomputers, Cost analysis, Data Processing, Data Processing Terminals, Programming Languages.

MINICOMPUTERS IN LIBRARY CIRCULATION AND CONTROL
by Lehmann, K.D.
Stadt- Und Universitaetsbibliothek, Frankfurt Am Main (West Germany)
December 1976
NASA 77N16933
The growing need for information services poses increasing problems for libraries and documentation centers. Data processing techniques provide several possibilities for improvement, among the most recent techniques are the use of minicomputers. A description of the mode of operation in library loan posting is given, as well as a discussion of the extent and structure of data, linkage possibilities, and special operational features. For these specialized applications a short survey of the hardware configuration and software of minicomputers is also presented. Comparison is made between a stand alone system and a minicomputer connected to a background computer. This study is based on actual projects existing in the Federal Republic of Germany.

Data Links, Information Dissemination, Libraries, Minicomputers, Computer programs, Germany, Information Systems, Management Methods.

THE USE OF A MINI-COMPUTER AT THE DEFENCE RESEARCH INFORMATION CENTRE (DRIC)
by Hart, G.W.
Defence Research Information Centre, Orpington (England)
December 1976

NASA 77N16932
The functions of the Defence Research Information Centre (DRIC) are outlined. A minicomputer is used to prepare the 'Abstracts Bulletin' and its indexes, and to provide data on the exchange of reports with foreign countries. Future possible applications for the computer are described. These include a register of the interests of DRIC's customers, a loans control system particularly for classified reports, thesaurus look up to help the scientific staff, and information retrieval (both SDI and Retrospective). A brief summary of other uses of computers in the UK Ministry of Defence Information and Library Services is included.

Defense Industry, Indexes (Documentation), Information Dissemination, Minicomputers, Abstracts, Computer Programs, Information Management, Man Machine Systems, Visual Aids.

ADVANCEMENTS IN RETRIEVAL TECHNOLOGY AS RELATED TO INFORMATION SYSTEMS
Advisory Group for Aerospace Research and Development, Paris (FRANCE)
December 1976
NASA 77N16930
Any present and future applications of computer technology to information management are explored.

Data Processing Equipment, Human Factors Engineering, Information Retrieval, Minicomputers, Telecommunication, Computer Storage Devices, Data Bases, Environmental Quality, Information Management.

IMPLEMENTATION OF A MINI-COMPUTER BASED INFORMATION SYSTEM. PART 1 SYSTEM OVERVIEW
by McIvor R.A.
Defence Scientific Information Service, Ottawa (Ontario)
August 1974
NASA 75N10725
An automated processing system for entering bibliographic and project data dealing with scientific and technical information, and for producing document digests, catalogue cards or microfilm cartridges, and Selective Dissemination of Information (SDI) notices, has been designed and implemented on a minicomputer system. A consolidated patron file was also implemented. The report describes in general terms the programmes which were developed and their interrelations. Examples of the outputs are included.

Information Systems, Minicomputers, Selective Dissemination of Information, Bibliographies, Computer Programming, Data Processing, Tables (Data)

MINICOMPUTERS A REVIEW OF CURRENT TECHNOLOGY, SYSTEMS, AND APPLICATIONS
by Hollingworth, Dennis
Santa Monica, Ca., RAND Corporation
1973
75V58182
NASA, Digital Computers

A PRACTICAL GUIDE TO MINICOMPUTER APPLICATIONS.
by Coury, Fred F.
IEEE Press New York, VI, 211 p. Illus. 28 CM.
1972
72V34025
NASA, Batch Processing, Computer Design, Computer Programs, Computer Techniques, Data Acquisition, Data Management, Data Processing, Data Processing Terminals, Digital Computers, Industries, Information Systems, Medical Science, Minicomputers, On-Line Programming, Research, Telecommunication, Time Sharing, Utilization

LIBRARY AUTOMATION PROJECT: FINAL REPORT TO OSTI ON PROJECT NUMBER S1/51/19, APRIL 1968-MARCH 1974
by Woods, R.G.
Southampton, University of Southampton.
1975
OSTI Report 5223

EFFECTIVE USE OF MINICOMPUTERS IN ON-LINE INTERACTIVE INFORMATION SERVICE SYSTEMS
By Tsuda, J.; Takeichi, N.; Hirano, C.
1st USA-Japan Computer Conference Proceedings, Tokyo, Japan;3-5 Oct 1972.
1972

IMPLEMENTATION OF A MINI-COMPUTER BASED INFORMATION SYSTEM. PART XVIII. A GENERAL FILE DUMP PROGRAM FOR THE VARIAN 620 COMPUTER
by Plunkett, Gordon W.
AD a008 519
Minicomputers, Information Systems, Computer Programming, Mathematical Logic, Debugging (Computers), Canada

RAPID GENERALIZED MINICOMPUTER TEXT SEARCH SYSTEM INCORPORATING ALGEBRAIC ENTRY OF BOOLEAN STRATEGIES
by Isenhour, T.L.; Woodward, W.S.; Lowry, S.R.
J Chem Inf Comput Sci V.15 N.2
May 1975

(9 Lecture series.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
14	AGARD-LS-92	ISBN 92-835-1276-6	UNCLASSIFIED

5. Originator

Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly sur Seine, France

6. Title

6 THE APPLICATION OF INEXPENSIVE MINICOMPUTERS TO INFORMATION WORK.

7. Presented

on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

8. Author(s)		9. Date
Various	12 89p.	11 Mar[redacted]78

10. Author's Address	11. Pages
Various	94

12. Distribution Statement

This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.

13. Keywords/Descriptors

Computers	Information retrieval	Publishing
Miniaturization	Editing	Libraries
Information centres	Abstracts	

14. Abstract

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

This ~~Lecture~~ report Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI (Selective Dissemination of Information) and retrospective retrieval and library housekeeping.

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD.

400 043 Gm

B-3

AGARD Lecture Series No.92
Advisory Group for Aerospace Research and Development, NATO
THE APPLICATION OF INEXPENSIVE MINI-COMPUTERS TO INFORMATION WORK
Published March 1978
94 pages

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

P.T.O.

AGARD-LS-92

Computers
Miniaturization
Information centres
Information retrieval
Editing
Abstracts
Publishing
Libraries

AGARD Lecture Series No.92
Advisory Group for Aerospace Research and Development, NATO
THE APPLICATION OF INEXPENSIVE MINI-COMPUTERS TO INFORMATION WORK
Published March 1978
94 pages

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

P.T.O.

AGARD-LS-92

Computers
Miniaturization
Information centres
Information retrieval
Editing
Abstracts
Publishing
Libraries

AGARD Lecture Series No.92
Advisory Group for Aerospace Research and Development, NATO
THE APPLICATION OF INEXPENSIVE MINI-COMPUTERS TO INFORMATION WORK
Published March 1978
94 pages

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

P.T.O.

AGARD-LS-92

Computers
Miniaturization
Information centres
Information retrieval
Editing
Abstracts
Publishing
Libraries

AGARD Lecture Series No.92
Advisory Group for Aerospace Research and Development, NATO
THE APPLICATION OF INEXPENSIVE MINI-COMPUTERS TO INFORMATION WORK
Published March 1978
94 pages

Minicomputers are now extremely powerful and can be equipped with large access stores. These features make them ideally suited to information work and their cost is sufficiently low that an information centre or service can even justify having one solely for its own use. This avoids all the problems inherent in the sharing of a main frame computer, either in an associated organization or at a commercial bureau.

P.T.O.

AGARD-LS-92

Computers
Miniaturization
Information centres
Information retrieval
Editing
Abstracts
Publishing
Libraries

This Lecture Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI and retrospective retrieval and library housekeeping.

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD, presented on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

ISBN 92-835-1276-6

This Lecture Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI and retrospective retrieval and library housekeeping.

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD, presented on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

ISBN 92-835-1276-6

This Lecture Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI and retrospective retrieval and library housekeeping.

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD, presented on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

ISBN 92-835-1276-6

This Lecture Series outlines the ways in which many computers can be used in information work and includes examples of their current use in a number of different areas, such as editing and publishing information bulletins, SDI and retrospective retrieval and library housekeeping.

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Technical Information Panel and the Consultant and Exchange Programme of AGARD, presented on 17–18 April 1978 in Delft, Netherlands and 20–21 April 1978 in Ankara, Turkey.

ISBN 92-835-1276-6